SAMANEH KHOSHROU

Learning In Evolving Video Streams

Ph.D. Thesis

FACULDADE DE ENGENHARIA DA UNIVERSIDADE DO PORTO
August 2016

FACULDADE DE ENGENHARIA DA UNIVERSIDADE DO PORTO

Learning In Evolving Video Streams

Samaneh Khoshrou

MAP-TELE Programme

Supervisor: Jaime dos Santos Cardoso (PhD)

Co-supervisor: Luís Filipe Teixeira (PhD)

August 2016

Abstract

With the advent of video surveillance networks, making sense of an ever-increasing amount of visual data is a growing desire and need in many applications. The automatic (re-)identification of individuals across camera views, which is known as *person re-identification (ReID)*, has been one of the most challenging problems in the field of video analysis. Current algorithms assume that a one time training session in where labelled observations of all the classes to be observed is available. Furthermore, the visual streams are recorded during a limited time frame. These assumptions are a good approximation to some of the real-life problems, however, in many of the applications some of the previous assumptions are violated, rendering the conventional algorithms suboptimal or impractical.

In this work, we look at the problem of human re-identification from a time-series mining perspective where visual streams are generated endlessly and initially we only have a vague knowledge of what concepts can be observed during the surveillance period. We formalize a learning concept suitable for multi-camera video surveillance and propose a learning methodology adapted to that new paradigm. We build a framework that resorts to evolving approaches, specifically ensembles, to track the environment online, and adapt the knowledge to accommodate the environment changes accordingly. The framework is also designed to exploit active learning strategies, in order to interact wisely with oracles, requesting assistance in more ambiguous to classify observations.

First, we utilize conventional discriminative ensembles to learn from streams generated in wide-area surveillance. The framework is then extended with *class-based ensembles*, where an individual ensemble is trained for every single class. Despite significant improvement in performance, stability in high-dimensional visual data and finding a robust novel class detection criterion are still the main concerns. Therefore, we improve the framework resorting to the universal background models to robustly learn individual object models from small samples and to more effectively detect novel classes. However, the framework may fall into the pitfall of the growing computational complexity, when faced with never-ending streams. To avoid the effect of stability-placity dilemma without sacrificing and possibly enhancing the performance, an intelligent learning algorithm is introduced that wisely develops over time with respect to drift level in order to reflect the latest concepts. The framework emphasises on coverage and accuracy as well as efficiency. We also propose a context-sensitive solo representation for video fragments which employs a novel unsupervised criterion to give more credit to more "trustworthy" observations.

In the numerous experiments that are reported, experimental evidence of the effectiveness of the learning framework and the representation scheme is provided.

Resumo

Com o crescimento dos sistemas de vigilância de vídeo, muitas aplicações precisam de fazer sentido de uma quantidade, cada vez maior, de dados visuais.

(Re-)identificar automaticamente indivíduos ao longo de várias câmaras, conhecido por "person re-identification" (ReID), tem sido um dos maiores desafios na área de análise de vídeo.

Os atuais algoritmos assumem que está disponível uma sessão de treino para um período de tempo em que todas as observações estão anotadas para todas as classes que serão observadas. Além disso, o sistema é esperado operar apenas durante período de tempo limitados .

Estas suposições são uma boa aproximação para alguns dos problemas reais. Contudo, em várias aplicações, algumas destas suposições são violadas, tornando os algoritmos convencionais subótimos ou inaplicáveis. Neste trabalho, olhamos para o problema da re-identificação duma perspetiva de aprendizagem duma série-temporal onde os streams visuais são gerados sem fim e inicialmente possuímos apenas conhecimento vago de quais conceitos poderão ser observados durante o período de vigilância.

Formalizamos um conceito de aprendizagem apropriado para vigilância de vídeo com várias câmaras e propomos uma metodologia de aprendizagem adaptada a esse novo paradigma. Construímos uma framework que recorre a abordagens que podem evoluir, especificamente a ensembles, para fazer o tracking de forma online, e adaptar o conhecimento de forma a acomodar mudanças no ambiente. O framework também está desenhado para explorar estratégias de aprendizagem ativa, de forma a interagir sensivelmente com o operador, pedindo ajuda nas observações de classificação mais ambígua. Primeiro, utilizamos ensembles discriminativos convencionais para aprender a partir de streams gerados em vigilância de zonas amplas. O framework é estendido com ensembles baseados em classes, onde cada ensemble individual é treinado para cada classe individual. Apesar de melhoramentos significativos no desempenho, a estabilidade em dados visuais de grande dimensão e o critério de deteção robusta de novas classes são ainda preocupações relevantes. Portanto, propomos um framework que recorre ao modelo de universal background para aprender de forma robusta modelos para cada objeto a partir de pequenas amostras e para detetar classes novas de forma mais eficiente. Contudo, o framework pode sofrer de complexidade computacional crescente, quando usado com streams contínuos. Para evitar o efeito do dilema estabilidade-plasticidade sem sacrificar e possivelmente até melhorar o desempenho, um algoritmo de aprendizagem inteligente foi introduzido que se desenvolve ao longo do tempo no que diz respeito ao nível de drift de forma a refletir os conceitos mais recentes. O framework enfatiza a amplitude e precisão da classificação, assim como a eficiência. Também propomos uma representação sensível ao contexto de fragmentos de vídeos que emprega uma critério não-supervisionado de forma a dar maior reconhecimento às observações de "maior confiança".

Nas inúmeras experiências que são relatadas, é fornecido evidência da eficácia do sistema de quadro de aprendizagem e de representação.

To Hassan

and

To Mina *and* Ali *, my devoted parents*

Acknowledgement

My deepest gratitude and warmest thanks go to my supervisor Professor Jaime S. Cardoso. I am extremely thankful and indebted to him for sharing his knowledge, supporting me throughout this journey with his expertise, constant presence, and patience whilst allowing me to find my way. I am very thankful to my co-supervisor Professor Luís F. Teixeira for his inspiring ideas. Thank you so much!

Next, I need to thank all the people who have created such a warm and friendly atmosphere at INESC-TEC, specially all the past and current members of *Visual Computing and Machine Intelligence* group for their friendship and support. I also owe a huge debt to Dr. Simon Malinowski for his helpful advices and fruitful discussions. A sincere thanks goes to Ricardo Cruz for kind help with Portuguese translation.

At École de technologie supérieure (Université du Québec), I had the honour to work with Prof. Eric Granger and his amazing team. Thank you so much for valuable suggestions and amazing discussions.

Last, but by no means least, my eternal gratitude goes to my family, without whom I would not be here. To my parents for their unconditional love and support throughout my life. Thank you for helping me through every step of my life , and for instilling in me the will to pursue my dreams. To my husband, Hassan, who has been a source of love, inspiration, and encouragement to me. Thank you for being the best friend and companion through every stage of this journey. To my sister Hannaneh and my brothers Majid and Hamid, whose love and constant support have always been heart warming through my cold moments.

"Let the beauty of what you love, be what you do"

Rumi

Contents

List of Figures

List of Tables

Acronyms

GMM	Gaussian Mixture Models
SIFT	Scale Invariant Feature Transform
IE	Intra-Ensemble
MCE	Micro-Ensemble
NEVIL	Never Ending Visual Information Learning
BCL	Batch Confidence Level
GMM	Gaussian Mixture Models
UBM	Universal Background Model
MML	Modified Margin Level
MC	Most Confident Level
BoW	Bag of Words
RoI	Region of Interest
FK	Fisher Kernels
IFK	Improved Fisher Kernel
MLB	Manually Labelled Batches
TB	Total Batches
SVM	Support Vector Machines
MAP	Maximum A Posteriori
IDSM	Individual Specific Model

Chapter 1

Introduction

1.1 Relevance and Problem Definition

With the advent of video surveillance networks, making sense of an ever-increasing amount of visual data is a growing desire and need in many applications. The automatic (re-)identification of individuals across camera views, which is known as *person re-identification (ReID)*, is a challenging and widely studied problem in this area [69,99,141] and it underpins many crucial applications such as long-term multi-camera tracking [55], behaviour analysis, and security monitoring.

With few exceptions, most state-of-art approaches to person ReID have focused on settings in which the persons of interest are known beforehand, and the system runs for a limited period of time by matching the pairs of small shots or images. This matching approach is clearly limited due to the drifting nature of the appearance of individuals and possible similarity of different individuals appearances, thus exploring spatio-temporal information from video sequences seems an interesting field to be studied. Although spatial-temporal information has been extensively explored in other video analysis task (i.e. activity recognition [59]), its use for person ReID is much less explored. Furthermore, in a real-world surveillance network, cameras capture streams of visual data endlessly.

In video surveillance environments, the underlying distribution of data changes over time - often referred to as *concept drift* - either due to intrinsic changes (pose change, movement, etc.), or extrinsic changes (lighting condition, dynamic background, complex object background, changes in camera angle, etc.). Thus, models need to be continually updated to represent the latest concepts. Moreover, when new objects enter the scene - referred to as *class evolution* [95] in machine learning literature - new models need to be trained for the novel classes. The problem gets further complex when the system is faced with *unbounded streams* of data [11].

Figure 1.1 demonstrates a typical surveillance scenario. Depending on the view angle and the quality of the camera, every surveillance camera covers an area called Field of View (FoV). When entering the scene, the object will enter the coverage area of at least one of the cameras. In such environments where objects move around and cross the FoV of multiple cameras, it is more than likely to have multiple streams, potentially overlapping in time, recorded at different starting

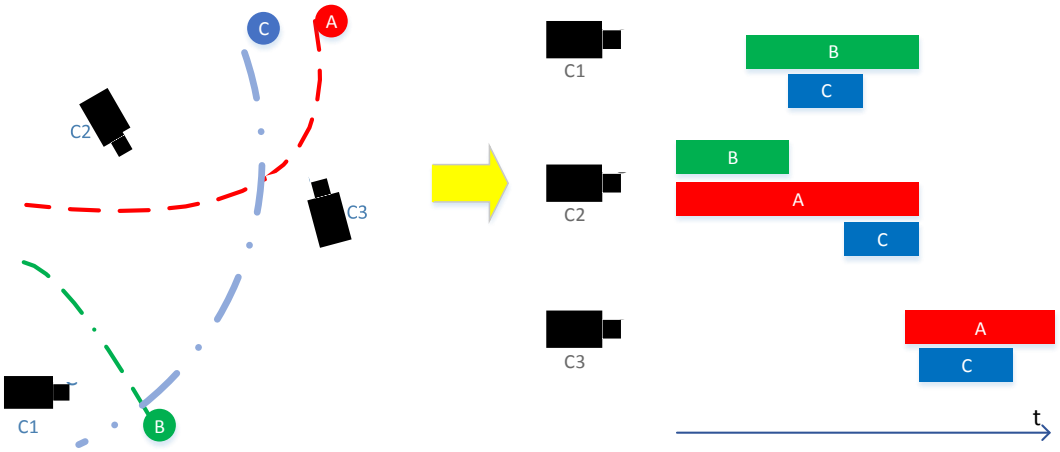


Figure 1.1: Typical surveillance scenario [75]

points with various lengths, for the same individual object (Figure 1.1). The surveillance system will have to track that object from the first moment it was captured by a camera and across all cameras whose fields of view overlap the object's path. Thus, a suitable outcome of the framework could be a timeline graph assigning each stream in each camera to an identity for the indicated presence period, as illustrated in Figure 1.2. This graph can be used for behaviour analysis as well as security purposes. In this simple scenario the typical tracking systems are likely to encounter some problems. In fact, mutual occlusion may occur if persons B and C cross. Consequently, their identities can be switched. Moreover, prolonged occlusion might occur, which might lead to track loss or mistaken identities [136]. Since the cameras are supposed to track all objects in their coverage area, the definition of a global identity for each object is necessary. Multiple appearances of objects captured by the same or by different cameras are identified in the process, allowing also to know the path followed by a given object. This setting is inherently different from person re-identification scenarios, either image-to-image [46] or video-to-video [108], that seek to determine if the images (videos) correspond to the person(s) of interest [139]. Whereas, this framework focuses on the design of a system, where no pre-defined class of interest is available. Moreover, typical person re-identification works assume that the acquired data has enough detail to support identification on facial data, while in our setting appearance-based approaches are more likely to be successful. We especially focus on long-term tracking, where people cross the FoV of multiple cameras over time and cameras monitor the environment for unbounded period.

Learning in such scenario can be characterized as follows:

Definition: Let υ be a set of unregulated time-series υ_i. Streams are potentially with concept drift as well as concept evolution. Each observation x within each stream is in a d-dimensional space, $x \in R^d$. Recording is not limited to a bounded period.

Requirements: An effective and appropriate one-pass algorithm to fit in our scenario is required to: a) learn from multiple unregulated streams; b) handle multi (possibly) high-dimensional

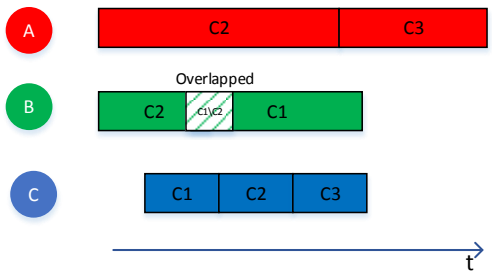


Figure 1.2: Desired outcome. A, B, and C represent different individuals; C1, C2, and C3 represent different cameras covering the scene.

data; c) handle concept drift; d) accommodate new classes; e) deal with massive unlabelled data; f) be of limited complexity.

Considerable body of multi-camera surveillance research assumes that adjacent camera view overlap [57, 63, 82, 142], whereas [64, 65, 98, 110, 124] assume non-overlapping views. Our proposed method makes no assumption of overlapping or non-overlapping views; hence, it can be applied in either settings.

1.2 Main Contributions

In this thesis, we put forward a framework to learn continuously from parallel video streams with partially labelled data and that allow us to learn novel knowledge, reinforce existing knowledge that is still relevant, and forget what may no longer be relevant. The kind of on-line learning approach may suffer if labelling errors accumulate, which are inevitable. Unrelated objects will sooner or later be assigned the same label or different labels will be assigned to different views of the same object. To help mitigate this issue, we will allow the system to interact with an oracle to help it stay "on track". The framework utilizes active learning methodologies appropriate for this problem. The framework receives directly the tracked sequences outputted by the tracking system and maintains a global object identity common to all the cameras in the system.

- Initially an ensemble of *discriminative classifiers*, NEVIL, is proposed in order to actively classify parallel video streams. The framework learns a classifier as new data is observed.

- The framework has been then substantially altered using a class-based ensemble strategy, in where an individual model is learnt for each class. The framework utilizes *generative models* to train the classifiers, allowing a double threshold strategy to detect novel classes.

  In a first step, the most popular group of generative models, *Gaussian Mixture Models*, are employed to learn new knowledge in NEVIL.gmm.

  The framework is further extended in NEVIL.ubm using generative models built by the maximum a-posteriori (MAP) adaptation of the *universal background model (UBM)*. The

adopted approach enables us to achieve a good balance between the need to have enough data to make reliable decisions and the need to adapt quickly enough to drifts and new concepts in the data streams.

- Analysing large volumes of visual information confronts vision researchers with the problem of finding effective methods to describe the objects that compose the captured scenes. The representation schemes used for this task need to effectively discriminate objects but at the same time they need to be efficient in terms of computational and storage resources. In a non-stationary environment these schemes should also accommodate changes in the objects‘ appearances. In this task we focus on how perpetual learning can be supported from the visual representation point of view. In contrast with the abstract models studied in the learning task, we will look more closely to data. The goal is to find the primitives that are more appropriate for updating and storage.

- Additionally, due to the video acquisition process, the quality of captured data may drift in time. Some works [131] have shown that better quality leads to more accurate models. Since low quality data may make future decisions highly ambiguous, it is crucial to select "good" frames from which a model is learnt for long-term tracking. Exploring the effect of observations' quality, as a contributing factor, on the performance of unseen target ReID systems seems a noteworthy area which has not been investigated before. Here we put forward an algorithm that gives more weight to "better" Region of Interests (RoIs) to build a model.

- We propose Intelligent NEVIL (INEVIL), a new learning framework, which is specially designed for long-term tracking of previously unseen objects over multiple cameras. Inspired by never-ending learning approaches, we employ active (detect & re-act) techniques to control the complexity of the most popular group of passive approaches, ensemble based models [42], in a time evolving environment.

  The active approach is based on a change detection strategy that triggers an adaptation with respect to the level of drift by updating or building a classifier. We assess the effectiveness of change detectors at both data and model level.

  - We put forward a data-level change detection mechanism that inspects features extracted from batches of RoIs. A simple yet effective video batch change criterion is introduced to measure the divergence between the current batch of RoIs and the reference one.

  - The Model-level mechanism is designed to react to gradual and recurrent drift by updating a classifier. The adaptation process is achieved through the merging of two GMMs into a single one, which has not been addressed before.

1.3 Document Structure

Chapter 2 reviews the background and related work relevant to this thesis. The Discriminative ensembles are introduced in Chapter 3. We provide a detailed presentation of the Class-based ensembles in Section 4, starting with an overview of NEVIL.gmm and then filling in the details of NEVIL.ubm. We discuss exploiting fusion approaches to provide a solo representation per batch in Chapter 5. In Chapter 6 we propose a context-sensitive batch representation. An intelligent framework for unsupervised long-term tracking of objects is presented in Chapter 7. Finally, conclusions are drawn in Chapter 8.

1.4 Achievements

Chapter 3 of this dissertation has been published as Samaneh Khoshrou, Jaime S. Cardoso, Luís F. Teixeira, "Active Mining of Parallel Video Streams", CoRR abs/1405.3382, 2014.

Section 4.1 has been published as Samaneh Khoshrou, Jaime S. Cardoso, Luís F. Teixeira, "Active Learning from Video Streams in a Multi-camera Scenario", 22nd International Conference on Pattern Recognition (ICPR), Stockholm, pp. 1248-1253, 2014.

Section 4.2 has been published as Samaneh Khoshrou, Jaime S. Cardoso, Luís F. Teixeira, "Learning from evolving video streams in a multi-camera scenario", Machine Learning, 100(2-3), 609-633, 2015.

Chapter 5 of this dissertation has been published as Samaneh Khoshrou, Jaime S. Cardoso, Eric Grnger, Luís F. Teixeira, "Spatio-Temporal Fusion for Learning of Regions of Interests Over Multiple Video Streams", 11th International Symposium on Visual Computing (ISVC), 509-520, 2015.

Chapter 6 and Chapter 7 have been submitted for publication as Samaneh Khoshrou, Jaime S. Cardoso, Eric Grnger, Luís F. Teixeira, "Unsupervised Long-Term Monitoring of Targets Over Multiple Video Cameras", 2016.

We also participated in some national conferences with the following papers:

- Samaneh Khoshrou, Jaime S. Cardoso, Luís F. Teixeira, " Learning in Evolving Video", In Proceedings of the 1st PhD. Students Conference in Electrical and Computer Engineering, Porto, Portugal, 28 - 29 June, 2012.

- Samaneh Khoshrou, Jaime S. Cardoso, Luís F. Teixeira, "Evaluation of Different Incremental Learning Methods for Video Surveillance Scenarios", In Proceedings of the 18th Portuguese Conference on Pattern Recognition (RECPAD2012), Coimbra, Portugal, 26 October, 2012.

- Samaneh Khoshrou, Jaime S. Cardoso, Luís F. Teixeira, "Learning from Uneven Video Streams in a Multi-camera Scenario", In Proceedings of the 19th Portuguese Conference on Pattern Recognition (RECPAD2013), Lisbon, Portugal, 31 October, 2013.

- Samaneh Khoshrou, Jaime S. Cardoso, Luís F. Teixeira, "Toward Building an Automatic Video Surveillance System", In Proceedings of the 20th Portuguese Conference on Pattern Recognition (RECPAD2014), Covilhã, Portugal, 31 October, 2014.

Chapter 2

Literature Review

2.1 State of the Art

Intelligent video surveillance (IVS) is a multi-disciplinary field, related to computer vision, pattern recognition, signal processing, communication, embedded computing and image sensors [142]; however, much of the history of IVS systems has addressed the problem employing computer vision techniques [13, 82, 87, 98, 101, 154]. Most approaches have addressed person ReID by comparing spatial appearance features such as color or texture of a pair of shots per individuals [46, 62, 92, 93, 132, 152]. Considerable variation between shots of different views and visual ambiguity due to individual clothing similarity make these approaches less practical for real-word problems. Spatio-temporal approaches are categorized in four groups [141]: 1) Multi-shot person re-identification: multiple images have been exploited to train a model for every individual. Multiple shots have been used to enhance spatial features or extracting additional features [145]. 2) Space-time features: space-time descriptors are designed to provide a compact representation of video sequences either based on space-time interest points or space-time volume/patch. Space-time volume representations (e.g HOG3D [78]) extracts robust and rich descriptors mainly based on gradient information which makes this group a successful descriptor in the action recognition problem [88, 99]. 3) Gait recognition: some of the state of art algorithms reformulated person ReID as a gait recognition problem. Since such methods usually require an ideal setting (i.e. uncluttered background, consistent silhouette extraction, etc.) to provide effective representation [67, 83, 103, 104, 143], they may fail in a real-world scenario. 4) Temporal sequence matching: Image sequences have been used to perform direct sequence matching. Simonnet et al. [127] used dynamic time warping to perform temporal sequence matching. Since the system requires regulated streams for accurate matching, specifically when strong noise is available, the framework may fail in real-world applications. Wang et al. [141] proposed an algorithm that selects discriminative fragments to learn a video ranking function, without implicit assumption on sequence alignment. Similar to the most of the state of art approaches, Wang's method assumes a *closed-world setting* where the test set comprises exactly the same individuals as the gallery set. The re-identification problem becomes further challenging when the test set contains mostly

Method	Parallel Streams	Uneven Streams	Concept Drift	Class Evolution	Learning	Complexity	Data
[107, 144]	×	×	√	×	SL	Constrained	MD
[2]	×	×	√	×	SL	Unconstrained	MD
[39,40,44]	×	×	√	√	SL	Unconstrained	MD
[97]	×	×	√	√	SSL	Constrained	MD
[18, 39, 156]	×	×	√	×	SSL	Constrained	MD
[79]	×	×	√	×	Clustering	Unconstrained	MD
[96]	×	×	√	√	Clustering	Constrained	MD
[14, 27, 29, 116]	√	×	√	√	Clustering	Constrained	1D

Table 2.1: Assessment of learning methods. √ and × denote being fit and inappropriate for the purpose, respectively. "*MD*" and "*1D*" denote multi-dimensional and one-dimensional data. "SL" and "SSL" indicate Supervised Learning and Semi-Supervised Learning.

non-target individuals [153].

Here, we look at the problem as learning from multiple data streams (visual data) in wild environments, that views segments of a stream as a unique element to classify, thus single stream mining methods cannot be employed. Learning from time-changing data streams has mostly appeared in the data mining context and various approaches have been proposed [51, 70]. Most of the methods proposed for parallel stream mining [14, 27, 29] require equal-length streams coming from a fixed number of sources. Thus, they would fail to leverage information from time-varying video tracks.

Learning in such non stationary environment requires evolving approaches that can adapt to accommodate the changes accordingly. The adaptation problem has been addressed by either active or passive approaches [42]. The active approach is designed to detect concept drift in order to trigger an adaptation [7, 50], whereas the passive one continuously update the knowledge every time new data is received [40]. While active approaches are more effective in online settings with abrupt drift, passive approaches are better suited for batch learning in settings with gradual drift

and recurrent concepts [42].

Ensemble based approaches are the most popular group of passive methods due to higher accuracy, flexibility and stability to handle concept drift [2, 79] and in some recent works class evolution [44], as well. Learn++.NSE [44] is one of the latest ensemble-based classification methods in literature, that generates a classifier using each batch of training data and applies a dynamic weighting strategy to define the share of each ensemble in the overall decision. As success is heavily dependent on labelled data, this method would not be applicable in wild scenarios. ActMiner [96] addresses the problem of concept-drift as well as concept evolution in a single infinite length data stream. Masud in [97] proposed an online clustering algorithm for single stream that employs an active strategy in order to minimize oracle collaboration. COMPOSE [43] is designed for learning from non-stationary environment facing gradual drift but it cannot support neither abrupt drift nor class evolution. Although some works [18, 39, 156] can handle more dramatic changes in data distributions, novel concept detection is an issue.

Additionally, these approaches may fall into the pitfall of a growing computational complexity, when faced with never-ending streams. To avoid the effect of stability-placity dilemma without sacrificing and possibly enhancing the performance [94, 158], some algorithms maintain a fixed size ensemble, by removing the oldest [16, 129] or the least contributing member [79, 90]. These techniques have mostly focused on two idealized settings: 1) Both training and test data are drawn from the same yet unknown distributions and the prediction of the ensemble can be evaluated on the training set at the first place [151]. 2) the data generating process produces sequence(s) of data in time from potentially different probability distributions and concepts. The prediction of the ensemble will be verified with the labels arriving immediately with the next chunk of data or after a fair delay (in verification latency scenarios) [128]. Thus, they will fail in scenarios in where labelled data is acute. Table 2.1 presents a qualitative look at the extent to which the reviewed methods fulfil the requirements for deployment in our scenario.

The Never-Ending Language Learning (NELL) [20] research project has been the inspiration of numerous researches to address the never-ending learning problem [28, 58]. Obviously, the techniques used by research works are informed by different assumptions in respect with the applications and goals. With a few exceptions [148], most of the never-ending literature has focused on coverage of knowledge, while our approach tries to cover knowledge and accuracy as well as efficiency.

2.2 Datasets

Video analysis in unconstrained environment has become a very important issue for many applications. To stimulate research in this field many datasets have been introduced. We classified these sets into two main groups: 1) synthetic sets. 2) real video clips.

Drift Rate	**C1**				**C2**				**C3**				**C4**			
	μ_x	μ_y	δ_x	δ_y	μ_x	μ_y	δ_x	δ_y	μ_x	μ_y	δ_x	δ_y	μ_x	μ_y	δ_x	δ_y
$0 < r < 0.25$	2	5	$0.5r$	$0.5+2r$	$5-5r$	8	$3-10r$	1	$5-5r$	2	$0.5+10r$	0.5	8	$8+15r$	0.5	0.5
$0.25 < r < 0.5$	–	–	–	–	15	$-1+5r$	1	$2+3r$	$1-4r$	2	0.5	$3-4r$	$5-5r$	13	$0.25+4r$	$0.5+4r$
$0.5 < r < 0.75$	10	$-10+5r$	1	$2-r$	17	$2+5r$	0.25	0.15	-1	$-2-4r$	0.25	0.15	–	–	–	–
$0.75 < r < 1$	–	–	–	–	20	$4r$	1	2	-7	$-5-r$	$7+4r$	$1+4r$	–	–	–	–
Drift Rate	**C5**				**C6**				**C7**							
	μ_x	μ_y	δ_x	δ_y	μ_x	μ_y	δ_x	δ_y	μ_x	μ_y	δ_x	δ_y				
$0 < r < 0.25$	12	15	2	$2+2r$	-15	$-5+15r$	1	$2+3r$	10	$5r$	0.5	$2+3r$				
$0.75 < r < 1$		–	–	–	–	–	–	–	$--10$	$-1+5r$	r	$2+3r$				

Table 2.2: Parametric Equations for classes of MS dataset

Dataset	No. of Streams	Range	No. Classes	Imbalance Degree	No. of Cameras	Setting
OneLeaveShopReenter1	3	[85 – 160]	2	0.28	2	Overlapped
OneLeaveShopReenter2	3	[63 – 347]	2	0.11	2	Overlapped
WalkByShop1front	6	[40 – 225]	4	0.22	2	Overlapped
EnterExitCrossingPaths1	6	[34 – 216]	4	0.23	2	Overlapped
OneStopEnter2	7	[51 – 657]	4	0.19	2	Overlapped
OneShopOneWait1	10	[36 – 605]	4	0.25	2	Overlapped
OneStopMoveEnter1	42	[10 – 555]	14	0.14	2	Overlapped
PETS2009	19	[85 – 576]	10	0.13	2	Overlapped
SAIVT-SOFTBIO	33	[21 – 211]	11	0.12	8	Overlpped, Nonoverlpped
SAIVT-NonOver	14	[21 – 211]	7	0.12	2	Nonoverlpped
SAIVT-Recurrent	28	[63 – 633]	7	0.65	2	Nonoverlpped

Table 2.3: The datasets characteristics. Imbalance degree [102] is defined by the ratio of sample size of minority class to that of the majority ones ; Range is defined by the length of shortest and longest streams in a given dataset, respectively.

2.2.1 Synthetic Dataset

Numerous synthetic datasets have been proposed in the literature [44, 97]. The synthetic dataset is generated in the form of (X, y), where X is a multi-dimensional feature vector, drawn from a Gaussian distribution N(μ_X, δ_X), and y is the class label. Since in real applications visual data may suffer from both gradual and abrupt drift, both situations are simulated by changing μ_X and δ_X in the parametric equations.

In this thesis, we employed a process that is similar to the one used in [44]. Table 2.2 presents these parametric equations applied for drift simulation; we generated 7 classes $(C1, C2, ..., C7)$; for some $(C5, C6)$ data changes gradually while others also experience one $(C1, C4, C7)$, or three $(C2, C3)$ dramatic drifts. The dataset was organized in 4 different scenarios with different levels of complexity, including streams with gradual drift, sudden drift, re-appearance of objects and non-stationary environments where we have abrupt class and concept drift. Each scenario includes:

- *Scenario* I: gradually drifting streams of 5 classes.
- *Scenario* II: streams with abrupt drifts of 5 classes.
- *Scenario* III : re-appearance of objects.
- *Scenario* IV: a non-stationary environment with class evolution as well as concept drift.

These scenarios are depicted in Fig. 2.1.

2.2.2 Real Video Clips

Many data sets targeting visual surveillance scenarios such as person detection and tracking, event detection, activity recognition, inter-camera tracking and re-identification, have been published over the years. Here we review some important datasets for the task of person re-identification.

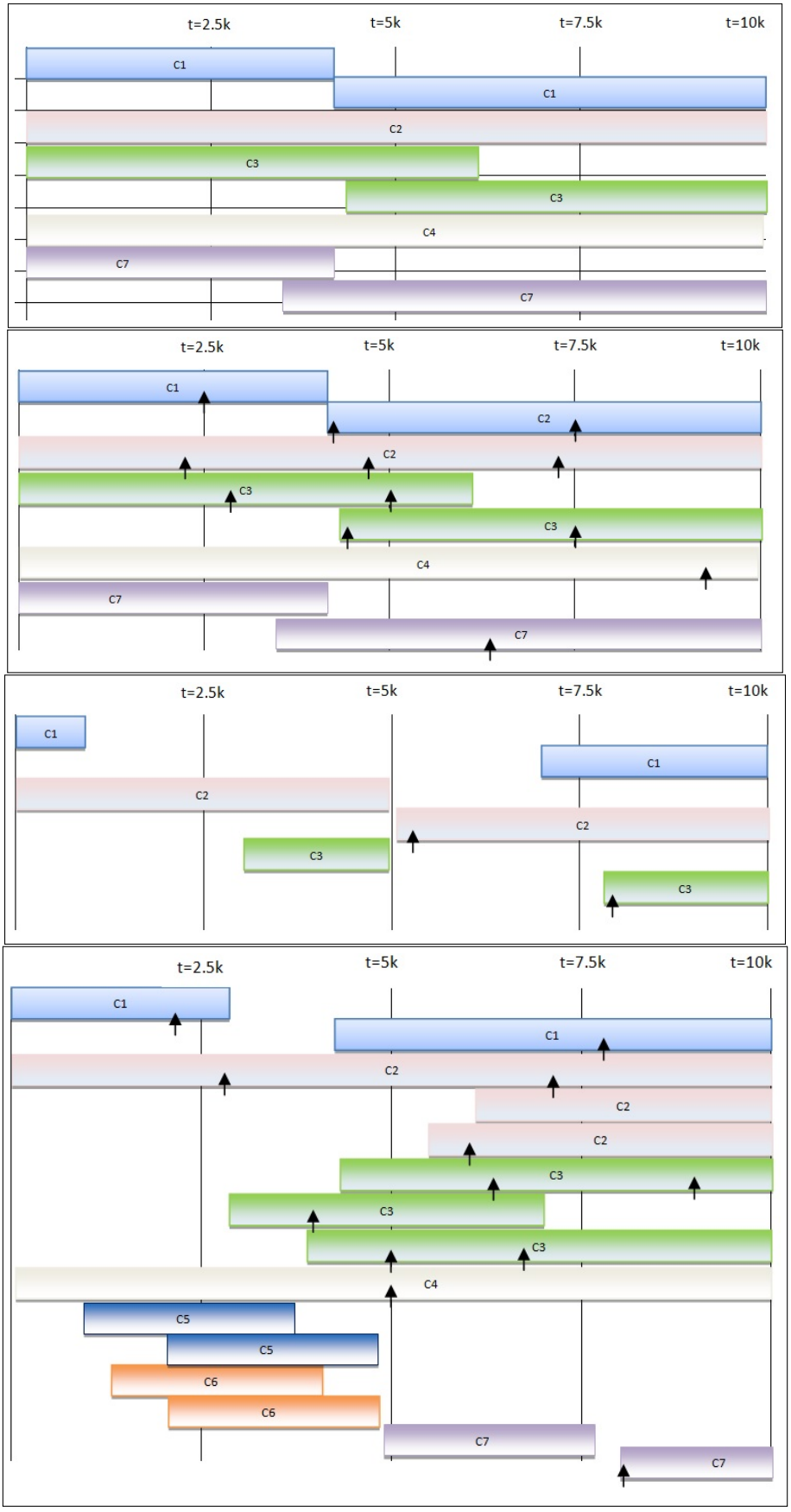


Figure 2.1: Scenarios in MS dataset. The sign ↑ denotes the occurrence of an abrupt drift in the nature of data.

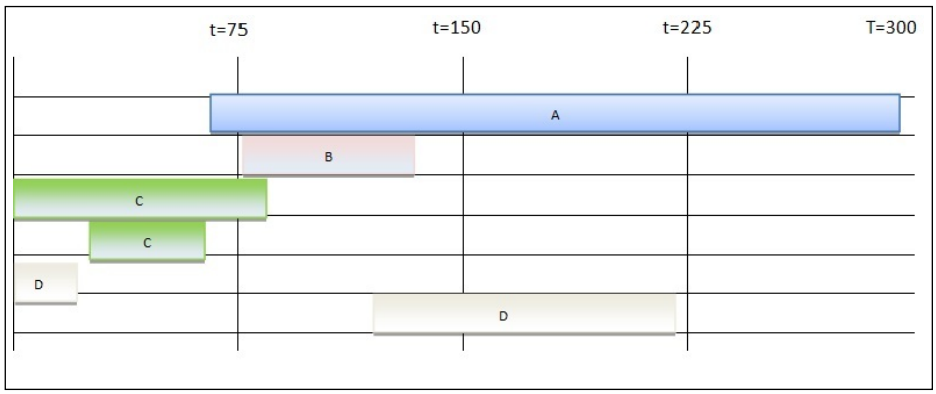


Figure 2.2: The EnterExitCrossingPaths1 scenario in the CAVIAR dataset. A, B, C, and D denote individuals who were present in the scene. (Note, labels do not carry any semantic information.)

MIT pedestrians data set

MIT pedestrians data set [106], introduced in 1997, to train pedestrian detectors. The dataset includes 924 frontal and rear views of pedestrian in single frames acquired from multiple video and photographic sources.

CAVIAR

CAVIAR dataset [112], which was introduced in 2004, was one of the first to provide video sequences instead of single frames. It was also the first to provide annotations of people location, identity and activity, making this data set useful for many problems. The CAVIAR first set was acquired with a single camera at INRIA Labs Grenoble for activity recognition, while the second version was recorded in a shopping mall in Lisbon using two cameras with overlapping fields of view, view of the corridor and frontal view, making it more interesting for RE-ID and multi-camera tracking. The sequences include people walking alone, meeting with others, window shopping, entering and exiting shops, fighting and passing out. As an explanatory sample, Fig. 2.2 depicts the streams in the *EnterExitCrossingPaths1* scenario.

PETS2009

PETS2009 [47] was recorded for the workshop at Whiteknights Campus, University of Reading, UK. The dataset comprises multi-sensor sequences containing crowd scenarios with three levels of complexity (i.e. subjective difficulty level/ density of the crowd). Dataset S2 addresses people tracking with 8256 frames. S2.L1 subclass exhibits a randomly walking sparse crowd.

SAVIT-SOFTBIO

SAVIT-SOFTBIO [15] is specially designed for the task of re-identification. This dataset consists of 150 sequences of 152 subjects travelling in a building environment though up to eight camera views, appearing from various angles and in varying illumination conditions. Each individual

is tracked from the moment he enters the building until he leaves the view of the surveillance network. The dataset contains 64,472 frames in total. The frames with occlusion are omitted.

In order to study the framework on different conditions, we build three subsets of SAVIT-SOFTBIO:

1) *SAVIT*: this dataset consists of 11 people travelling in a building environment through up to eight camera views, appearing from various angles and in varying illumination conditions reflecting real-world conditions. There are streams generated by both overlapped and non-overlapped views.

2) *SAVIT Non-Over*: this set includes streams captured by cameras with non-overlapping FoV. These sequences present challenging situations with cluttered scenes, different illumination conditions as well as different scales of the person being captured.

3) *SAVIT Recurrent*: We conducted an experiment on a subset of streams of seven subjects captured by non-overlapped cameras. Inspired by [86] and in order to simulate the recurring concept drifts with sufficient learning data, this set is organized periodically.

HDA+

The dataset [48] was acquired from 13 indoor cameras distributed over three floors of our research department, recording simultaneously for nearly 30 minutes. More than 64000 annotations were performed on a total of more than 75000 frames. The video recordings exhibit a high degree of variability in terms of image resolutions, frame rates, points of view. The different illumination conditions of the recording areas make the dataset challenging for tasks such as person detection and re-identification[1].

We wrap up this section in Table 2.3, presenting a qualitative look at the characteristics of the datasets applied in our work. Various factors have been considered in the table including: imbalance degree [102] that is defined by the ratio of sample size of minority class to that of the majority one; range in the length of streams that defines the length of shortest and longest streams in a given dataset, respectively.

To extract the RoIs, we employed an automatic tracking approach [135] to track objects in the scene and generate streams of bounding boxes, which define the tracked objects' positions. As the tracking method fails to perfectly track the targets, a stream may include RoIs of distinct objects.

2.3 Evaluation Criteria

Cumulative Matching Characteristic curve (CMC) [100] is the standard metric for person re-identification in a closed-world setting. The CMC curve represents the probability of finding the correct match over the first n ranks.

There is no metrics in person re-identification that can be used readily for an open-world verification task [153]. Thus we have to define a set of new ones.

[1]This dataset was made available in a late stage of this thesis; despite our genuine interest it was not possible to be applied in the experimental work.

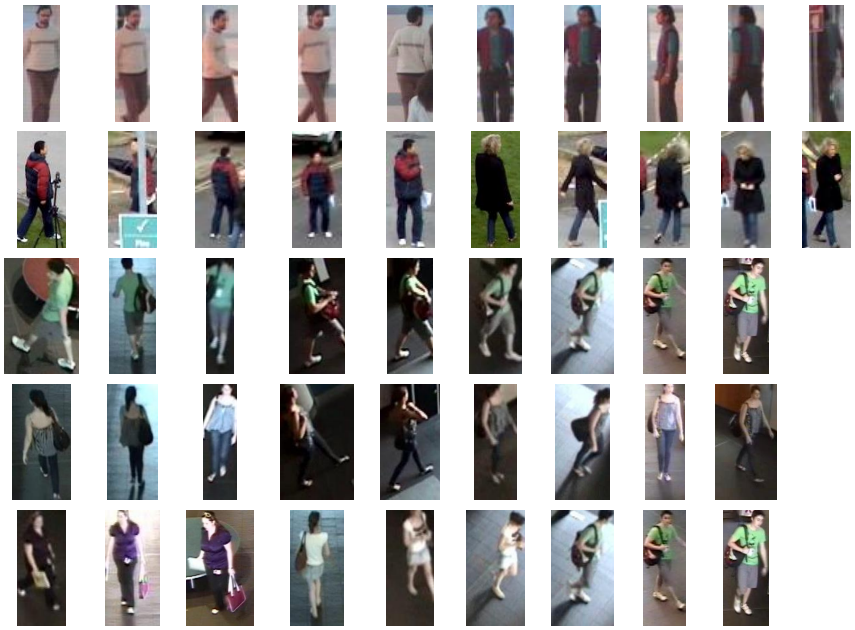


Figure 2.3: An example of diversity in appearance

In our framework, the key tactic to tackle open world re-identification is invoking teachers using active learning strategies which aims to achieve high accuracy using as little annotation effort as possible. Thus, a trade-off between accuracy and proportion of labelled data can be considered as one of the most informative measures.

Accuracy

In a classical classification problem the disparity between real and predicted labels explains how accurately the system works. However, in our scenario the labels do not carry any semantic meaning (it is not a person recognition problem). The same person should have the same label in different batches, whichever the label. One is just interested that, whatever label is used to represent a person, it is correctly transported to the next batches. The labels are therefore permutable and just define a partition of the set of all batches according to which label was assigned to it. As such, when evaluating the performance of our framework we are just comparing the partition of the set of batches as defined by the reference labelling with the partition obtained by the NEVIL labelling. We adopted a generic partition-distance method for assessing set partitions, initially proposed for assessing spatial segmentations of images and videos [19, 81]. Thus, the accuracy of the system is formulated as:

$$Accuracy = \frac{N - Cost}{N} \tag{2.1}$$

where N denotes the total number of batches, and $Cost$ refers to the cost, yielded by the assignment problem.

Annotation

Assume MLB and TB denote the manually labelled batches and all the batches available during a period (includes one or more time slots), respectively. The *Annotation Effort* is formulated as:

$$Annotation\ effort = \frac{\#MLB}{\#TB} \tag{2.2}$$

It is expected that the accuracy increases with the increase of the annotation effort.

Area Under the Learning Curve (ALC)

ALC [21] is a standard metric in active learning research that combines *accuracy* and *annotation effort* into a single measurement. The rationale behind the use of such metric is that there is not a single budget level that everyone agrees is the reasonable cost for a particular problem. Hence, ALC, which provides an average of accuracy over various budget levels, seems to be a more informative metric. Herein, the learning curve is the set of accuracy plotted as a function of their respective annotation effort, a, $Accuracy = f(a)$. The ALC is obtained by:

$$ALC = \int_0^1 f(a)da \tag{2.3}$$

2.4 Representation

The choice of visual features, or image representation, is a key choice in the design of any classic image classification system [24]. Seems fair to say that most of improvement in such system performance can be associated to the introduction of improved representation from the classic Bag of Visual Words (BoW) [31] to the Fisher Vector (FV) [109]. In such approaches, local image features are extracted. Then, the features are encoded in a high dimensional image representation.

Local features are properties of an image (object) located in a single point or small region. It is a single piece of information describing a rather simple, but ideally distinctive property of the object's projection to the camera (image of the object). Examples for local features of an object are the color, (mean) gradient or (mean) gray value of a pixel or small region. One of the main advantages of using point features is their resilience to partial object occlusions. When one part of an object is occluded, point features from the non-occluded part can still be used for the tracking.

Various local features have been proposed in the literature: *Scale-Invariant Feature Transform (SIFT)* [89] which is invariant to rotations, translations, scaling, affine transformations, and partially to illumination changes, *Speeded Up Robust Features (SURF)* [10], Fast Invariant to Rotation and Scale Transform (FIRST) [9], Local Binary Pattern (LBP) [105], Histogram of Oriented Gradient (HOG) [35], Rotation-Invariant Fast Feature (RIFF) [133], DAISY [137], Fast Retina Keypoint (FREAK) [5].

Encoding approaches A large number of encoding approaches for bag of visual words models have been proposed in the past few years to improve on the standard histogram of quantized local features. Examples include: histogram encoding, Kernel codebook encoding, Fisher encoding, super vector encoding, Locality-constrained linear encoding (LLC).

In general these approaches can be grouped into three categories: hard quantization, soft quantization, and Fisher method. Classical BoW computes a *spatial histogram (hard quantization)* of visual words constituting the baseline representation. Recent methods replace hard quantization with methods that retain more information. This can be done in two ways: 1) soft quantization or in other words, expressing the representation as combination of visual words (e.g. [136]), and 2) expressing the representation as the difference between the features and visual words (e.g. FV) [23].

Recently, there has been a renewed interest in Convolutional Neural Networks (CNNs) [84], comprising several layers of non-linear feature extractors [24]. Their sophisticated structure provide a deep representation of the data.

Chapter 3

Discriminative Ensembles

In this chapter we present the Never Ending Visual Information Learning (NEVIL) framework. NEVIL employs an ensemble of classifiers that are incrementally trained (with no access to previous data) on incoming batches of data, and combined with a form of weighted majority voting. NEVIL is designed for non-stationary environments in which no labelled data is available but the learning algorithm is able to interactively query the user to obtain the desired outputs at carefully chosen data points.

3.1 Never Ending Visual Information Learning

A high-level sketch of the proposed method is shown in Figure 3.1. A typical tracking algorithm analyses sequential video frames and outputs the movement of targets between the frames, generating multiple streams of visual data. Environmental challenges such as varying illumination, lack of contrast, bad positioning of acquisition devices, blurring caused by motion as well as occlusion make data often noisy and/or partially missing. We address these challenges by a batch divisive strategy, as learning from a data batch may reduce the noise and fill the gaps caused by miss-tracking.

The algorithm is provided with a series of data batches $\mathscr{D}_t^{m_i}$, where m_i is the index of the i-th stream present at time slot t, TS_t, (not all streams are necessarily present). Note that a stream corresponds to a track generated by the tracking system and a single camera can yield multiple streams. A single batch aggregates B frames. The starting time of each stream is potentially different from stream to stream but batches are aligned between streams. Inside each frame the data corresponds to some pre-selected object representation (e.g. bag of words, histogram) is extracted.

The ensemble obtained by all models generated up to the current time slot TS_t is named the composite hypothesis H_{t-1}. With the arrival of the current data batches $\mathscr{D}_t^{m_i}, \quad i = 1 \cdots M$, NEVIL tries to predict the class label for each of the batches in current TS_t based on the probability estimate $p(C_k|\mathscr{D}_t^{m_i}, H_{t-1})$, where C_k runs over all the class labels observed so far.

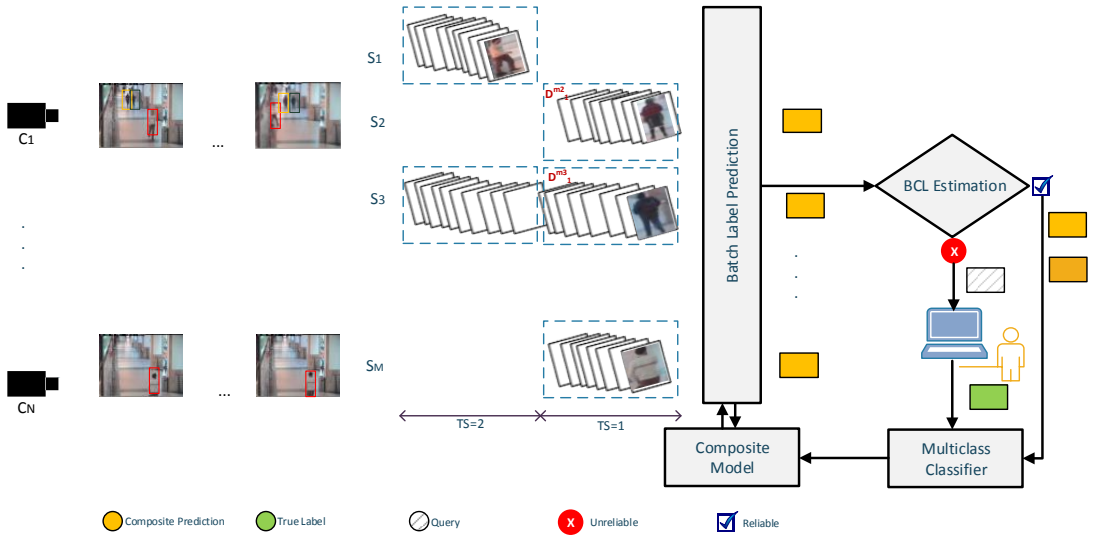


Figure 3.1: NEVIL High-level Overview

This kind of on-line learning approach can suffer if labelling errors accumulate, which is inevitable. Unrelated objects will sooner or later be assigned the same label or different labels will be assigned to different views of the same object. To help mitigate this issue, we allow the system to interact with a human, to help it stay on track.

Algorithm 1 outlines our approach. Initially, the composite model is initialized to yield the same probability to every possible class (uniform prior). When the batches $\mathscr{D}_1^{m_t}$ in time slot t become available, NEVIL starts with computing the probabilities $p(C_k|\mathscr{D}_t^{m_i}, H_{t-1})$ for each batch $\mathscr{D}_t^{m_i}$ in the time slot. Once $p(C_k|\mathscr{D}_t^{m_i}, H_{t-1})$ is obtained, a batch confidence label (BCL) is estimated; if BCL is high enough (above a prespecified threshold), the predicted label

$$\arg\max_{C_k} p(C_k|\mathscr{D}_t^{m_i}, H_{t-1})$$

is accepted as correct, otherwise the user is requested to label the data batch. The labelled batches (either automatically or manually) are used to generate a new multiclass classifier that is integrated in the composite model, yielding H_t.

Four tasks need now to be detailed: a) the batch label prediction (by the composite model); b) the BCL estimation; c) the multiclass classifier design in current time slot; d) the composite model structure and update.

3.1.1 Batch Label Prediction

A batch $\mathscr{D}_t^{m_t}$ is a temporal sequence of frames $\mathscr{D}_{t,f}^{m_t}$, where f runs over 1 to the batch size B. The composite model, H_{t-1}, can be used to predict directly $p(C_k|\mathscr{D}_{t,f}^{m_i}, H_{t-1})$ but not $p(C_k|\mathscr{D}_t^{m_i}, H_{t-1})$.

Algorithm 1 NEVIL

Input: $\mathscr{D}_t^{m_i}, i = 1, ..., M$
$W_0 \leftarrow \frac{1}{k}$
$H_0 \leftarrow W_0$
while $\mathscr{D}_t$ is $True$ **do**
    Batch label prediction (Section 3.1.1)
    $p(C_k|\mathscr{D}_t^{m_i}) \leftarrow (\mathscr{D}_t^{m_i}, H_{t-1})$
    Batch Confidence Level Estimation (Section 3.1.2)
    $BCL \leftarrow p(C_k|\mathscr{D}_t^{m_i}, H_{t-1})$
    Multiclass classifier design (Section 3.1.3)
    $h_t \leftarrow \mathscr{D}_t$
    Composite model structure and update (Section 3.1.4)
    $H_t \leftarrow (h_t,\ H_{t-1},\ W_t)$
end while

The batch (multiframe) Bayesian inference requires conditional independence

$$
\begin{aligned}
&p(\mathscr{D}_t^{m_i}|C_k, H_{t-1}) = \\
&p(\mathscr{D}_{t,1}^{m_i}, \cdots, \mathscr{D}_{t,B}^{m_i}|C_k, H_{t-1}) = \\
&p(\mathscr{D}_{t,1}^{m_i}|C_k, H_{t-1}) \cdots p(\mathscr{D}_{t,B}^{m_i}|C_k, H_{t-1}) = \\
&\textstyle\prod_{j=1}^{B} p(\mathscr{D}_{t,j}^{m_i}|C_k, H_{t-1})
\end{aligned}
$$

From there, and assuming equal prior probabilities, it is trivial to conclude that

$$p(C_k|\mathscr{D}_t^{m_i}, H_{t-1}) = Z \prod_{j=1}^{B} p(C_k|\mathscr{D}_{t,j}^{m_i}, H_{t-1}), \tag{3.1}$$

where Z is a normalization constant. In practice, products of many small probabilities can lead to numerical underflow problems, and so it is convenient to work with the logarithm of the distribution. The logarithm is a monotonic function, so that if $p(C_k|\mathscr{D}_t^{m_i}, H_{t-1}) > p(C_\ell|\mathscr{D}_t^{m_i}, H_{t-1})$ then

$$\log p(C_k|\mathscr{D}_t^{m_i}, H_{t-1}) > \log p(C_\ell|\mathscr{D}_t^{m_i}, H_{t-1}).$$

Then we can rewrite the decision as choosing the class that maximizes

$$\log p(C_k|\mathscr{D}_t^{m_i}, H_{t-1}) = \log Z + \sum_{j=1}^{B} \log p(C_k|\mathscr{D}_{t,j}^{m_i}, H_{t-1}) \tag{3.2}$$

The batch label prediction can also be analysed as a problem of combining information from multiple (B) classification decisions. Considering that, per frame, the composite model produces approximations to the a posteriori probabilities of each class, different combination rules can be considered to build the batch prediction from the individual frame predictions [6, 77]. While Eq. (3.1) turns out to be the product rule (or geometric mean), the sum rule (or arithmetic mean) is

also often preferred:

$$p(C_k|\mathscr{D}_t^{m_i}, H_{t-1}) = Z \sum_{j=1}^{B} p(C_k|\mathscr{D}_{t,j}^{m_i}, H_{t-1}) \tag{3.3}$$

In fact some authors have shown that the arithmetic mean outperforms the geometric mean in the presence of strong noise [6, 77]. Experimentally, we will compare both options.

3.1.2 The Batch Confidence Level Estimation

Having predicted a class label for a data batch, one needs to decide if the automatic prediction is reliable and accepted or rather a manual labelling should be requested.

Various criteria have been introduced as uncertainty measures in the literature for a probabilistic framework [123]. Perhaps the simplest and most commonly used criterion relies on the probability of the most confident class, defining the confidence level as

$$\max_{C_k} p(C_k|\mathscr{D}_t^{m_i}, H_{t-1}). \tag{3.4}$$

However, this criterion only considers information about the most probable label. Thus, it effectively "throws away" information about the remaining label distribution [123].

To correct for this, an option is to adopt a margin confidence measure based on the first and second most probable class labels under the model:

$$p(C^*|\mathscr{D}_t^{m_i}, H_{t-1}) - p(C_*|\mathscr{D}_t^{m_i}, H_{t-1}), \tag{3.5}$$

where C^* and C_* are the first and second most probable class labels, respectively. Intuitively, batches with large margins are easy, since the classifier has little doubt in differentiating between the two most likely class labels. Batches with small margins are more ambiguous, thus knowing the true label would help the model discriminate more effectively between them [123].

Note that while the estimation of the wining class for batch label prediction requires only the comparison of the relative values as given by Eq. (3.1), Eq. (3.2) or Eq. (3.3) both approaches Eq. (3.4) and Eq. (3.5), for the confidence level require the exact computation of the a posteriori probabilities of the classes. This involves computing the normalizing constant associated with Eq. (3.1) or Eq. (3.3), which is specially unstable for Eq. (3.1).

We therefore put forward variants of the two previous measures. As an alternative to the margin confidence measure Eq. (3.5), we base the confidence level on the *ratio* of the first and second most probable class labels:

$$BCL = p(C^*|\mathscr{D}_t^{m_i}, H_{t-1}) / p(C_*|\mathscr{D}_t^{m_i}, H_{t-1}), \tag{3.6}$$

which can be directly applied for the sum rule or modified to $\log p(C^*|\mathscr{D}_t^{m_i}, H_{t-1}) - \log p(C_*|\mathscr{D}_t^{m_i}, H_{t-1})$ for the product rule. Either way, we eliminate the issue with the normalization constant.

To come up with an alternative to the most confident class measure, we write the decision as

$$\max p_k = \frac{\prod_{j=1}^{B} p_{k,j}}{\sum_k \prod_{j=1}^{B} p_{k,j}} \gtrless T, \tag{3.7}$$

where we introduced the following simplifications in notation: $p_k = p(C_k|\mathscr{D}_t^{m_i}, H_{t-1})$ and $p_{k,j} = p(C_k|\mathscr{D}_{t,j}^{m_i}, H_{t-1})$. The comparison in Eq. (3.7) can be rewritten as

$$(1-T)\prod_{j=1}^{B} p_{k^*,j} \gtrless T \sum_{k,k\neq k^*} \prod_{j=1}^{B} p_{k,j}, \tag{3.8}$$

where $k^* = \arg\max_k p_k$. Since we cannot work directly with the log of Eq. (3.8) due to the sum in the denominator, we introduce the simplification of binarizing the classification in each frame, defining $\bar{p}_{k^*,j} = \sum_{k,k\neq k^*} p_{k,j} = 1 - p_{k^*,j}$.

Accepting the strong assumption of independence for the aggregated class, then

$$\bar{p}_{k^*} = \prod_{j=1}^{B} \bar{p}_{k^*,j}.$$

This ends up in exchanging the order of the sum and product in the right hand side of Eq. (3.8), which can now be rewritten as

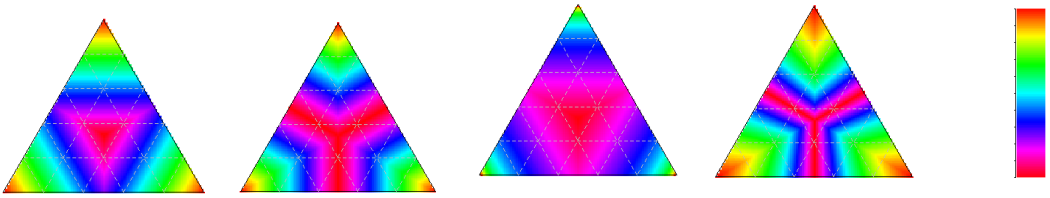
$$(1-T)\prod_{j=1}^{B} p_{k^*,j} \gtrless T \prod_{j=1}^{B} \bar{p}_{k^*,j}. \tag{3.9}$$

Now it is a trivial process to apply the log to obtain a stable decision:

$$\sum_{j=1}^{B} \log p_{k^*,j} \gtrless S + \sum_{j=1}^{B} \log \bar{p}_{k^*,j}, \tag{3.10}$$

where $S = \log T - \log(1-T)$.

Figure 3.2 highlights the characteristics of the four confidence measures by a ternary plot (where every corner indicates a class). This plot graphically depicts the ratios of the three variables (herein, occurrence of each class) as positions in an equilateral triangle. The probability of each class is 1 in its corner of the triangle. Moving inside triangle, the percentage of a specific class decreases linearly with increasing distance from the corner till dropping to 0 at the line opposite it. A rainbow-like color pattern shows the informativeness of different composition of three classes. For all methods, the least reliable batch would lie at the center of triangle, where the posterior label distribution is uniform and thus the least certain under the ensemble. Similarly, the most informative batch lies at the corners where one of the classes has the highest possible probability.



(a) Most confident class measure.

(b) Margin confidence measure.

(c) Modified most confident class measure.

(d) Modified margin confidence measure.

(e) Margin confidence measure.

Figure 3.2: Heatmaps illustrating the behavior of the reliability measures in a three-label classification problem.

3.1.3 Multiclass Classifier

At time slot t, we obtain a new set of batches that are automatically or manually labelled. We assume all the frames belonging to a batch are from the same object (and the underlying tracking system does not mix identities in the time slot period) and therefore the frames inside a batch correspond to observations of the same class. Consider that to the M batches in current time slot correspond $L < M$ labels (some batches can have the same label). We need the design a classifier that can approximate the a posteriori probability function $p(c_k|\mathscr{D}_{t,f}^{m_i})$, which gives the probability of the frame belonging to a given class c_k, given that $\mathscr{D}_{t,f}^{m_i}$ was observed.

A standard way to address this problem is to apply *discriminative approaches* which predict the conditional probability directly. As an alternative, *generative approaches* find the joint distribution $p(\mathscr{D}_{t,f}^{m_i}, c_k)$ and then use Bayes' rule to form the conditional distribution from the generative model. A third option is to find a function $f(\mathscr{D}_{t,f}^{m_i})$, called a discriminant function, which maps each input $\mathscr{D}_{t,f}^{m_i}$ directly onto a class label. In this case, and although probabilities play no role in the design of the discriminant function, it is still possible to get estimated for the conditional probabilities [17]. Each approach has its relative merits and we evaluate experimentally instantiations of each.

One of the challenges we need to handle in a practical scenario is when in a time slot all the batches have the same label (automatically or manually assigned). In these TSs the training of a multiclass classifier is not possible. We resort to one-class classifiers for these time slots, also known as unary classification, to distinguish the single class present in the training set (the batches in the time slot) from all other possible classes [71].

3.1.4 The Composite Model Structure and Update

The composite model H_t in the NEVIL framework is an ensemble of classifiers h_t that are incrementally trained (with no access to previous data) on incoming time slots of data as described previously. The individual models h_t are combined using a weighted majority voting, where the weights are dynamically updated with respect to the classifiers' time of design.

The prediction outputted by the composite model H_t for a given frame $\mathscr{D}_{t,f}^{m_i}$ is

$$p(C_k|\mathscr{D}_{t,f}^{m_i}, H_t) = \sum_{\ell=1}^{t} W_\ell^t h_\ell(C_K|\mathscr{D}_{t,f}^{m_i}),$$

where $h_\ell(.)$ is the multiclass classifier trained at TS ℓ, W_ℓ^t is the weight assigned to classifier ℓ, adjusted for time t.

The weights are updated and normalised at each time slot and chosen to give more credit to more recent knowledge. The weights are chosen from a geometric series $\frac{1}{p^t}, ..., \frac{1}{p^2}, \frac{1}{p}$, normalised by the sum of the series to provide proper probability estimates:

$$W_\ell^t = \frac{\frac{1}{p^{(t-\ell+1)}}}{\sum_{j=1}^{t} \frac{1}{p^j}}$$

3.2 Experimental Methodology

3.2.1 Experimental Setup

A series of experiments were conducted to explore the capabilities of the proposed framework. We evaluated the framework on synthetic datasets as well as *Caviar* clips. These sequences present challenging situations with cluttered scenes, high rates of occlusion, different illumination conditions as well as different scales of the person being captured.

We employ an automatic tracking approach [135] to track objects in the scene and generate streams of bounding boxes, which define the tracked objects' positions.

An hierarchical bag-of-visterms method is applied to represent the tracked objects, resulting in a descriptor vector of size 11110 for each frame (refer to [136] for additional details). In order to avoid the curse of dimensionality that system may suffer from, Principle Component Analysis (PCA) is applied to the full set of descriptor features as a pre-processing step. Hence, the number of features in each stream is reduced to 85 dimensions.

3.2.2 Instantiation of Classifiers

In Section 3.1.3, we identified three approaches that have been applied in the literature to obtain the posterior probability. A set of experiments were conducted in order to study the behaviour of our framework employing instances of each option. We chose the following methods: *Gaussian Mixture Models (GMM)* and *Naive Bayes* as *generative approaches*, *Support Vector Machines (SVM)* [22] as one of the most popular *discriminant function* and *logistic regression* [45] as a member of *discriminative approaches* family.

Designing a classifier for time slots where batches constitute different labels is quite straightforward. The challenging situation arises when we need to do unary classification. As we employed various approaches with specific characteristics, different strategies are proposed to handle this situation.

Single-class SVM classifies each frame as completely similar or different from given class, whereas generative approaches (GMM and Naive Bayes) provide the probabilistic estimation.

To the extent of our knowledge, using logistic regression in unary problems is an unexplored topic; existing methods need data generated by at least two classes in order to make the prediction. Therefore, we keep the batches from the last multi-class time slot and combine them with the uni-class time slot to build the training set.

3.2.3 Evaluation Criteria

Accuracy and Annotation effort have been applied to assess the performance of the framework on various datasets.

3.2.4 Baseline Methods

To the best of our knowledge, there is no method that mines multi-dimensional parallel streams in such a non-stationary environment, where the number and length of streams vary greatly. Therefore, we compare our framework with three baseline approaches:

- Passive Learning: The first half of all the batches are submitted to the oracle for labelling. Once the labelled set is obtained, a classifier is trained and applied to classify the other half of stream. This method is far from a real online active learning strategy, as it needs complete data available. For datasets in which there is no dramatic distribution evolution between first and second half, we expect that it provides an upper bound to be compared with our method.

- Even/Odd Learning: As an on-line baseline, for a given stream, batches are marked alternately with odd and even integers, where odd batches are kept in a buffer with their true labels. At each time slot, a model is re-trained using the buffer. We then use this model to classify even batches. Therefore, we may partly follow the distribution changes in this setting leading to better performance than Passive Learning. However, we need to keep all the odd batches, which is far from a practical solution in an on-line scenario.

- Unwise active learning: We use an unwise version of the original framework as a baseline, where all the batches occurred before initiation time (t_{int}) would be annotated. For $t > t_{int}$, NEVIL computes the probabilities of known classes. Once $p(C_k|\mathscr{D}_t^{m_i}, h_{t_{int}})$ are obtained, a batch confidence label (BCL) is estimated; if BCL is high (above a pre-defined threshold), the predicted label

$$\arg\max_{C_k} p(C_k|\mathscr{D}_t^{m_i}, h_{t_{int}})$$

is accepted as correct label of the batch, otherwise the user is requested to label the batch. The method is summarized in Algorithm 2. Despite meticulous selection of queries, as the model is not updated, the algorithm may establish a lower bound the level of performance that can be expected in an evaluation.

Table 3.1: Comparison of baseline approaches on multiple datasets

		Accuracy (%)	
Dataset	Multiple Classifier	Passive Learning	Even/odd Learning
ScenarioI	SVM	97.39	97.19
	GMM	79.61	79.45
	Naive bayes	79.61	79.45
	Logistic Regression	18.44	18.21
ScenarioII	SVM	66.32	72.25
	GMM	79.60	79.45
	Naive bayes	79.60	79.45
	Logistic Regression	40.85	35
ScenarioIII	SVM	74.70	76.65
	GMM	78.37	78.02
	Naive bayes	78.37	78.02
	Logistic Regression	62.18	62.64
ScenarioIV	SVM	80.05	78.61
	GMM	81.81	81.87
	Naive bayes	81.81	81.87
	Logistic Regression	45.51	40.65
EnterExitCrossingPaths1	SVM	89.28	93.7
	GMM	66.45	75.16
	Naive bayes	66.45	75.16
	Logistic Regression	80.12	79.86
OneLeaveShopReenter1	SVM	63.74	100
	GMM	64.06	61.49
	Naive bayes	64.06	61.49
	Logistic Regression	92.18	97.86
OneShopOneWait1	SVM	80.24	95.79
	GMM	92.33	52.88
	Naive bayes	92.33	52.88
	Logistic Regression	81.35	93.42
OneStopEnter2	SVM	83.56	98.63
	GMM	76.73	75.28
	Naive bayes	76.73	75.28
	Logistic Regression	81.36	93.02
WalkByShop1front	SVM	92.32	97.58
	GMM	91.50	90.93
	Naive bayes	91.50	90.93
	Logistic Regression	89.04	96.07
OneStopMoveEnter1	SVM	62.47	79.40
	GMM	90.99	90.93
	Naive bayes	90.99	90.93
	Logistic Regression	56.25	73.76

Algorithm 2 Unwise active learning

Input: $\mathscr{D}_t^{m_i}, i = 1, ..., M$
$h \leftarrow empty$
while $\mathscr{D}_t$ is $True$ **do**
  if $t > t_{int}$ **then**
    Batch label prediction
    $p(C_k|\mathscr{D}_t^{m_i}) \leftarrow (\mathscr{D}_t^{m_i}, h_{int})$
    Batch Confidence Level Estimation
    $BCL \leftarrow p(C_k|\mathscr{D}_t^{m_i}, h_{int})$
  else
    Multiclass classifier design
    $h_{int} \leftarrow \mathscr{D}_t$
  end if
end while

3.3 Results

Firstly, multiple tests were run to determine the optimal batch size for each dataset to be explored. Batch size was varied between 1% to 50% of the size of the longest stream available in each dataset. Experiments were repeated for 50 equally spaced values in that range. The optimal batch size varies and is influenced by the characteristics of the streams present in each dataset. Optimal batch sizes have been observed to range between 30 and 35 for real video streams and between 200 and 300 for synthetic sequences.

Table 3.1 provides a summary of the performance of *Passive Learning* and *Even/Odd Learning* using various classifiers on all Synthetic as well as Caviar video clips. Since different classifiers provide varying performances on different datasets, the need for a procedure that carefully assesses algorithms seems inevitable. We applied Friedman test [37] that provides a non-parametric rank based statistical significance test. This test is similar to parametric repeated measures ANOVA, which tests if there is a significant difference between the rank of different treatments across multiple attempts. When the test runs over all the datasets shows that null hypothesis is verified which means that type of the classifier has no significant effect on the overall performance of baseline method in real applications. However, the test shows that Logistic Regression has yielded weak results for synthetic data in both learning methods. When we perceive the superior learners based on the mean rank for various scenarios, generative approaches perform fairly better in the synthetic datasets, while discriminative methods win for real video clips. Since the dimension of real data is large, while synthetic data is generated in 2D space, these results also emphasizes the difficulties that generative models face in high-dimensional spaces. As mentioned in Section 3.2.4, we expect better or equal results from *Even/Odd Learning* than *Passive Learning* which is the case in all the settings applied discriminative approaches as well as almost all used generative methods. Unexpected behaviour of generative methods when applies on *OneShopOneWait1* dataset can be explained by high bias of these methods when trying to model such complex data.

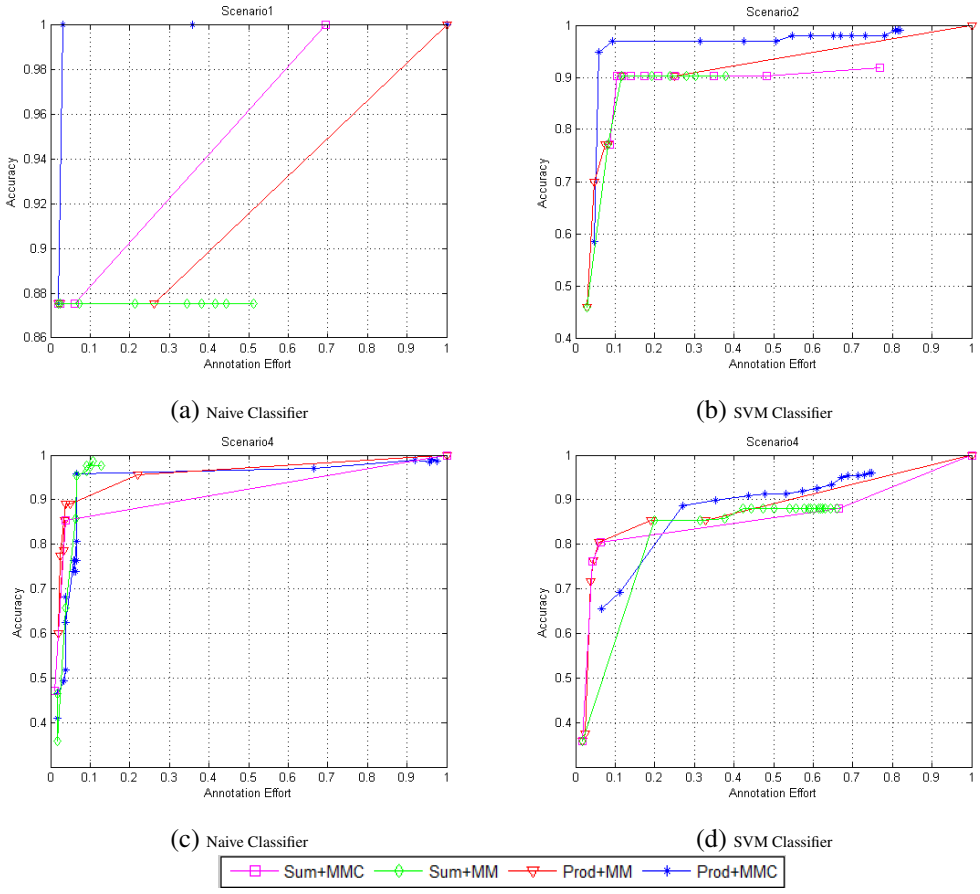


Figure 3.3: Multiple configurations tested on the synthetic scenarios. “SUM", “Prod", “MMC", and “MM" indicate sum rule, product rule, modified most confident and modified margin.

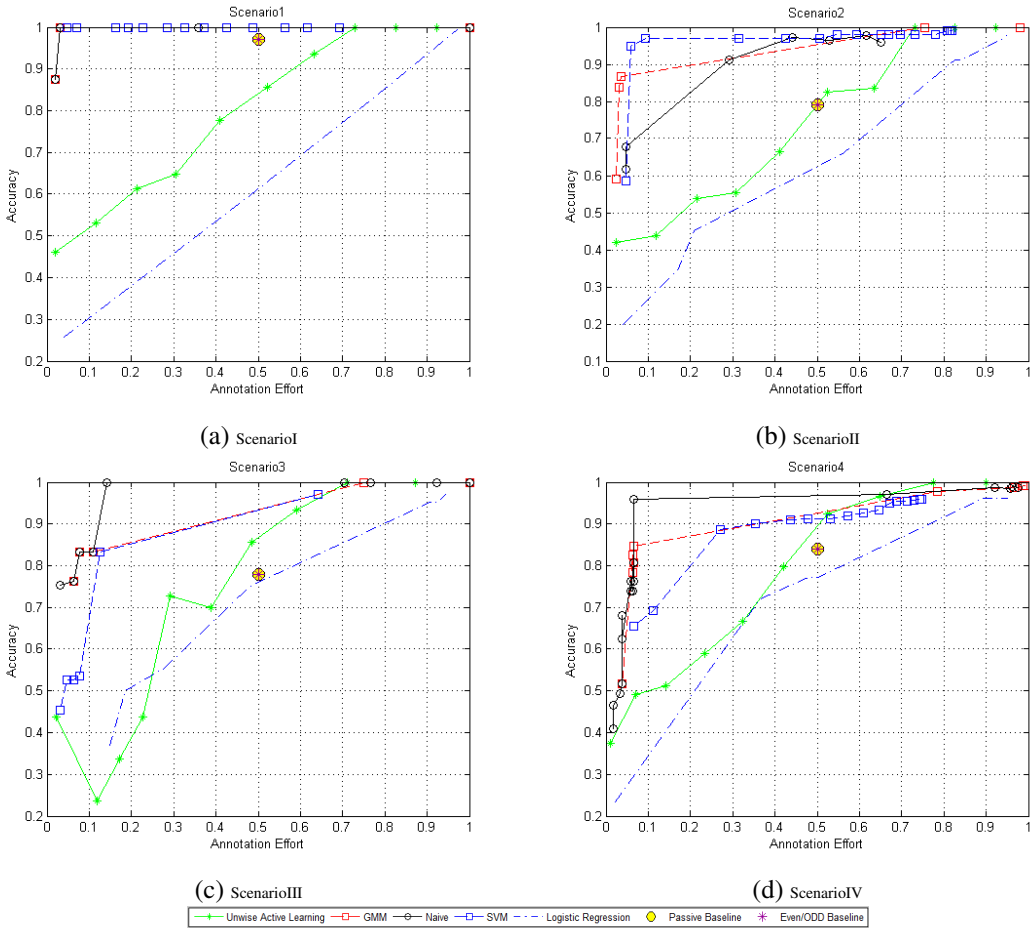


Figure 3.4: Multi-class classifier comparison on synthetic scenarios using the best configuration (Prod+MM).

Figure 3.3 presents the results of multiple settings on Scenarios I,..., IV. One prominent observation on all these results is that using geometric mean (Prod) to combine information of frames in a given batch and the modified most measure (MMC) to select most informative batches give the best performance.

Figure 3.4 illustrates the comparative results across multiple classifiers on *Scenarios* I,..., IV from which we can observe that: a) NEVIL achieves more than 90% accuracy with less than 15% annotation effort in all the datasets, which obviously outperforms baseline approaches. For all the sets, we reached equal accuracy to *Passive* as well as *Even/Odd* Learning while spending much less human resources. b) Naive classifier gives the best overall performance which emphasise the more flexible nature of generative models than discriminative ones. Needless to say, following the results depicted in Figure 3.3, we only present the result of winner setting.

Figure 3.5 shows the comparative results on some CAVIAR sequences with various NEVIL configurations. We observe that unlike synthetic scenarios, employing arithmetic mean (SUM) as combination method and modified margin (MM) as selection criteria present winner results. The presence of challenging noise in real data explains the different behaviour of the framework.

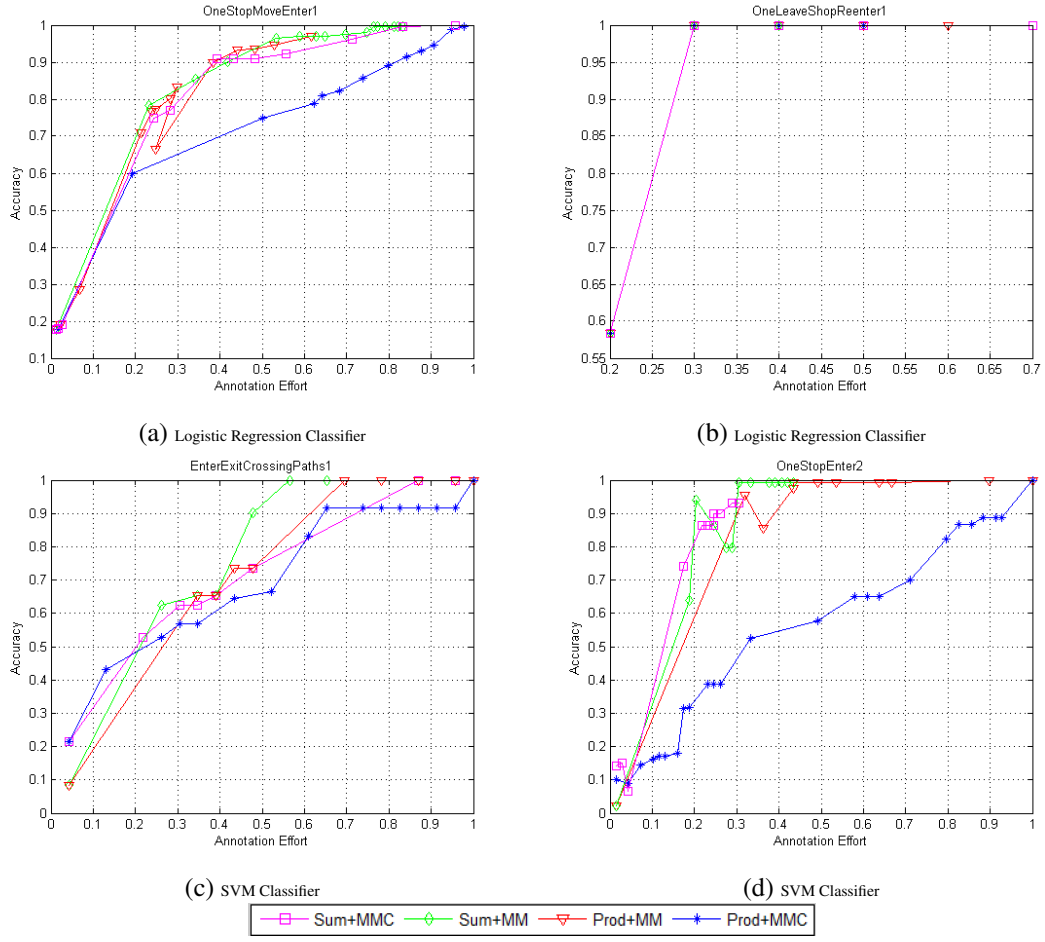
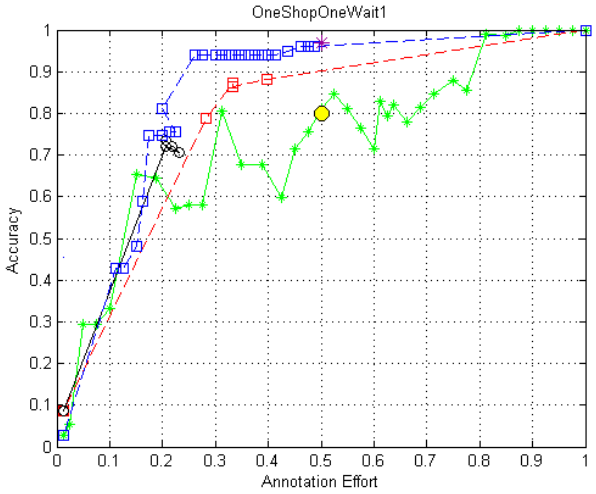
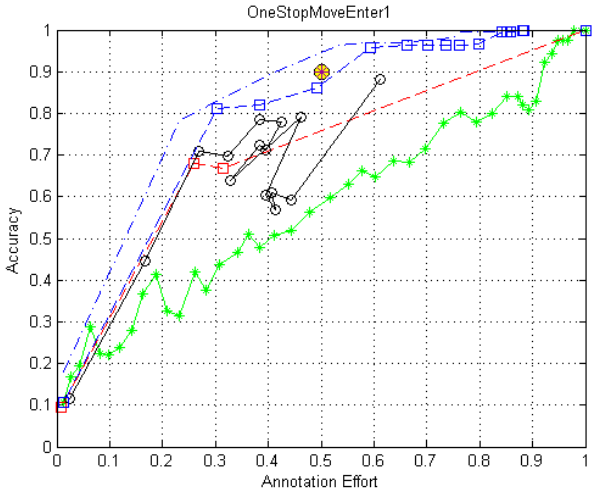


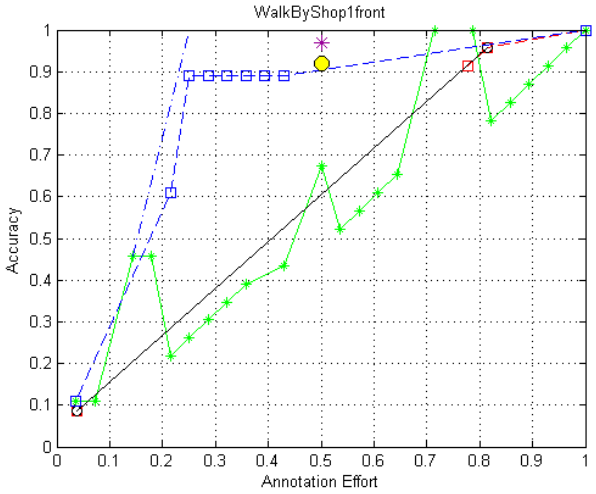
Figure 3.5: Multiple configurations tested on the CAVIAR sequences.“SUM", “Prod", “MMC", and “MM" indicate sum rule, product rule, modified most confident and modified margin.



(a) OneShopOneWait1



(b) OneStopMoveEnter1



(c) WalkByShop1front

Figure 3.6: Performance using multiple configurations on the CAVIAR sequences.

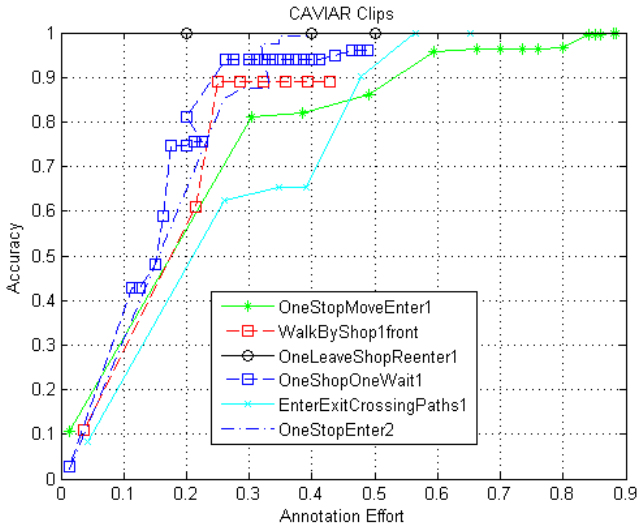


Figure 3.7: Performance of NEVIL on multiple CAVIAR sequences. The results were obtained with the SUM+MM configuration and applying SVM as the classifier.

Figure 3.6 presents the performance of NEVIL employing various classifiers on multiple CAVIAR sequences. The NEVIL framework achieves over 80% accuracy with less than 25% of labelling and in most cases, that is clearly superior to baseline methods. Contrary to results obtained from synthetic data, Discriminative models outperforms than Generative ones. Higher dimension of video streams (herein, equal to 85) may explain this behaviour. Generative models are commonly trained using Maximum-Likelihood Estimation (MLE) that especially for high dimensional data, the likelihood can have many local maxima. Thus, finding the global maximum affects the performance and renders the approach less practical.

Finally, Figure 3.7 presents the results obtained across multiple CAVIAR scenarios from the most successful setting, which means SVM, SUM, and MMC as base classifier, combination rule and selection criteria, respectively. Under such setting, NEVIL achieves 80% accuracy with 30% annotation effort for *OneStopMoveEnter1*, the most complex scenario with 42 streams from 14 classes.

3.4 Discussion

In this chapter, we address the problem of learning from visual streams generated in a multi-camera surveillance scenario using a discriminative ensemble.Inspired from active learning strategies, in our proposed framework (NEVIL) an oracle provides labelled batches; multiple informativeness measures are used to determine when the oracle is used. As base learners are bottlenecks of any learning pipeline, various groups of classifiers were studied and experimentally evaluated. We ran the experiments on synthetic as well as real datasets.

In synthetic scenarios, where low dimensional clean data is available, applying the geometric mean and the modified most confident measure gives the best and least expensive (in terms of annotation cost) results. However, to get the highest accuracy from noisy visual data we need to apply arithmetic mean for combining information and modified margin to select the most informative batches.

Another question we tried to answer was which classifier to use on a given dataset. In a low dimensional clean data, generative approaches give the best results, however obtaining robust and stable results from high dimensional data is too difficult, as shown by our experiments. The best performance is obtained through discriminative approaches.

Chapter 4

Class-based Ensembles

This work extends our prior framework, NEVIL [75], that was instantiated with *discriminative approaches* in order to actively classify parallel video streams. While having the benefit of a robust classifier design, this strategy is not without difficulties. Firstly, if a new class appears, it will be difficult to detect the novelty. Secondly, the framework becomes biased towards the majority class in the case of severe class imbalance. NEVIL trains a classifier for the batches available in a time slot. The classifiers are kept in an ensemble and participate in the final decision using a weighted sum strategy. If the decision is not reliable enough, the batch will be sent to an oracle to annotate it. Since NEVIL computes the posterior probability (that must sum to 1), it is likely to assign a high enough (reliable) probability to a new class and mislead the system even when exploiting *generative models*. Thus, NEVIL cannot exploit *generative models* true potential. All the aforementioned reasons make this method fairly expensive. Hence, we need to spend more human resources to get higher accuracy.

To address these problems, the NEVIL framework is extended with *generative models* within a new framework, allowing a double threshold strategy to detect novel classes and avoids the pitfall of classifying every batch as the majority class. A class-based ensemble is introduced, where models of each class are stored separately.

Class-based ensemble is firstly introduced in [4] where a model is trained for each class in a chunk. The ensemble keeps a fixed size ensemble of each class and it has been shown that this approach is more robust than traditional ensembles, though it needs the presence of training partition in each chunk. Although this approach does not fit directly in our scenario, we expect that class-based ensemble can improve our framework.

4.1 NEVIL.gmm

In this section we present our framework, that employs class-based ensembles to learn from non-stationary environments where no labelled data is available. The framework is able to interactively query the user to obtain the desired outputs at carefully chosen batches. The algorithm is a one-pass class-based ensemble of classifiers that trains a separate model (h_i^j) for a class (j) at every

time slot (t). It also keeps models of each class in a separate ensemble (*Intra-Ensemble*). As one of the most popular generative approaches, use of Gaussian Mixture Models (GMM) to train models seems a natural choice. A time-adjusted weighting strategy combines the probabilities outputted by the models in order to make the final decision.

The framework receives multiple visual streams, generated by a typical tracking algorithm, which analyses sequential video frames and outputs the movement of targets between the frames. Environmental challenges such as varying illumination, lack of contrast, bad positioning of acquisition devices, blurring caused by motion as well as occlusion make data often noisy and/or partially missing. We address these challenges by a batch divisive strategy, as learning from a data batch may reduce the noise and fill the gaps caused by miss-tracking.

4.1.1 Gaussian Mixture Models

The Gaussian mixture density is a weighted linear combination of M component densities. For a D-dimensional feature vector, x, the mixture density for class c is defined as:

$$p(x|c) = \sum_{i=1}^{M} \omega_i p_i(x) \tag{4.1}$$

where ω_i are the mixture weights which satisfy the constraint that $\sum_{i=1}^{M} \omega_i = 1$; $p_i(x)$ are the unimodel Gaussian density functions parametrized by a mean $D \times 1$ vector, μ_i, and a covariance $D \times D$ matrix, $\sum_i$;

$$p_i(x) = \frac{1}{(2\pi)^{D/2}|\sum_i|^{1/2}} exp\left\{-\frac{1}{2}(x-\mu_i)^T(\sum_i)^{-1}(x-\mu_i)\right\} \tag{4.2}$$

The parameters of application c's density model are denoted as $\theta_a = \{\omega_i, \mu_i, \sum_i\}$ where $i = 1, \cdots, M$. Such parameters are estimated via Expectation–Maximization (EM) algorithm [36], given a set of training vectors. The EM algorithm refines the GMM parameters through an iterative procedure aiming at maximizing the likelihood of the initialized model for the observed feature vectors.

4.1.2 Learning Framework

Similarly to NEVIL, the algorithm is provided with a series of data batches $\mathscr{D}_t^{m_i}$, where m_i is the index of the i_{th} stream present at time slot t, TS_t, (not all streams are necessarily present). Note that a stream corresponds to a track generated by the tracking system and a single camera can yield multiple unregulated streams. A single batch aggregates B frames. Inside each frame the data corresponds to some pre-selected object representation (e.g. bag of words, histogram).

The ensemble obtained by all *Intra-Ensemble (IE)* generated up to the current time slot TS_t is named the composite hypothesis H_{t-1}. With the arrival of the current data batches $\mathscr{D}_t^{m_i}, \quad i = 1 \cdots M$, the algorithm tries to predict the most probable class label for each of the batches in

Algorithm 3 NEVIL.gmm

Input: $\mathscr{D}_t^{m_i}, i = 1,...,M$
$W_0 \leftarrow \frac{1}{k}$
$H_0 \leftarrow W_0$
while $\mathscr{D}_t$ is $True$ **do**
  Batch label prediction (Section 4.1.2.1)
  $p(C_k|\mathscr{D}_t^{m_i}) \leftarrow (\mathscr{D}_t^{m_i}, H_{t-1})$
  Batch Confidence Level Estimation (Section 4.1.2.2)
  $BCL \leftarrow p(C_k|\mathscr{D}_t^{m_i}, H_{t-1})$
  Model design (Section 4.1.2.3)
  $h_t^j \leftarrow \mathscr{D}_t^j, \ \ j = 1,...,k$
  Composite model structure and update (Section 4.1.2.4)
  $IE_t^j \leftarrow h_t^j, \ \ j = 1,...,k$
  $H_t \leftarrow (IE_t^1,...,IE_t^k, \ H_{t-1}, \ W_t)$
end while

current TS_t based on the probability estimate $p(C_k|\mathscr{D}_t^{m_i}, H_{t-1})$, where C_k runs over all the class labels observed so far.

Algorithm 3 outlines our approach. Initially, the composite model is initialized to yield the same probability to every possible class (uniform prior). When the batches $\mathscr{D}_1^{m_t}$ in time slot t become available, the framework starts computing the probabilities $p(C_k|\mathscr{D}_t^{m_i}, H_{t-1})$ for each batch $\mathscr{D}_t^{m_i}$ in the time slot. Once $p(C_k|\mathscr{D}_t^{m_i}, H_{t-1})$ is obtained, a batch confidence label (BCL) is estimated; if BCL is high enough (above a predefined threshold), the predicted label

$$\arg\max_{C_k} p(C_k|\mathscr{D}_t^{m_i}, H_{t-1})$$

is accepted as correct, otherwise the user is requested to label the data batch. The labelled batches (either automatically or manually) are used to generate new separate models h_t^k (k runs over all the classes available in t) that are kept in intra ensembles IE_k, integrating in the composite, yielding H_t.

Four tasks need now to be detailed: a) the batch label prediction (by the composite model); b) the BCL estimation; c) the model design in current time slot; d) the composite model structure and update.

4.1.2.1 Batch Label Prediction

A batch $\mathscr{D}_t^{m_t}$ is a temporal sequence of frames $\mathscr{D}_{t,f}^{m_t}$, where f runs over 1 to the batch size B on top. The composite model, H_{t-1}, can be used to predict directly $p(C_k|\mathscr{D}_{t,f}^{m_i}, H_{t-1})$ but not

$p(C_k|\mathscr{D}_t^{m_i}, H_{t-1})$. The batch (multiframe) Bayesian inference requires conditional independence

$$\begin{aligned}
&p(\mathscr{D}_t^{m_i}|C_k, H_{t-1}) = \\
&p(\mathscr{D}_{t,1}^{m_i}, \cdots, \mathscr{D}_{t,B}^{m_i}|C_k, H_{t-1}) = \\
&p(\mathscr{D}_{t,1}^{m_i}|C_k, H_{t-1}) \cdots p(\mathscr{D}_{t,B}^{m_i}|C_k, H_{t-1}) = \\
&\prod_{j=1}^{B} p(\mathscr{D}_{t,j}^{m_i}|C_k, H_{t-1})
\end{aligned}$$

From there, and assuming equal prior probabilities, it is trivial to conclude that

$$p(C_k|\mathscr{D}_t^{m_i}, H_{t-1}) = Z \prod_{j=1}^{B} p(C_k|\mathscr{D}_{t,j}^{m_i}, H_{t-1}), \tag{4.3}$$

where Z is a normalization constant. In practice, as streams have different starting points and lengths, the number of frames may vary ($<= B$) for different batches in a given timeslot. Thus, the products of different number of small probability can make the process of defining an optimal confidence level threshold challenging. Thus, besides the options we considered in Chapter 3, we estimate the probability of a given batch by:

$$p(C_k|\mathscr{D}_t^{m_i}, H_{t-1}) = \mathscr{M}edian(p(C_k|\mathscr{D}_{t,j}^{m_i}, H_{t-1})) \tag{4.4}$$

That makes the prediction independent of the number of frames. Then, the framework will assign each batch to the class that maximizes $p(C_k|\mathscr{D}_t^{m_i}, H_{t-1})$.

4.1.2.2 The Batch Confidence Level Estimation

Having predicted a class label for a data batch, one needs to decide if the automatic prediction is reliable and accepted or rather a manual labelling be requested.

Various criteria have been mentioned as an uncertainty measure in chapter 3 for a probabilistic framework. Perhaps the simplest and most commonly used criterion relies on the probability of the most confident class, defining the confidence level as

$$\max_{C_k} p(C_k|\mathscr{D}_t^{m_i}, H_{t-1}). \tag{4.5}$$

4.1.2.3 Model Design

At time slot t, we obtain a new set of batches that are automatically or manually labelled. A model (h_t^j) has been trained by positive instances ($\mathscr{D}_t^j$) of classes at t. Hence, we require models that can carry deep information about the distribution of an individual class. Having in mind our needs, *generative models*, seem a natural choice. In this work, we use GMMs to learn the model for individuals at each TS. It is worth mentioning that we assume that all the frames belonging to a batch are from the same object.

4.1.2.4 The Composite Model Structure and Update

The composite model H_t is an ensemble of Intra-ensembles ($IE_t^j, j = 1,...,k$). Each IE_t^j includes models that are incrementally trained (with no access to previous data) on incoming batches of j^{th} class (h_t^j). The approximation outputted by individual models are combined using a weighted majority voting, where the weights are dynamically updated with respect to the classifiers' time of design.

The prediction outputted by the composite model IE_t^j for a given frame $\mathscr{D}_{t,f}^{m_i}$ is

$$p(C_k|\mathscr{D}_{t,f}^{m_i}, IE_t^j) = \sum_{\ell=1}^{t} W_\ell^t h_\ell^j(C_K|\mathscr{D}_{t,f}^{m_i}),$$

where $h_\ell^j(.)$ is the model trained from batches of j_{th} at TS ℓ, W_ℓ^t is the weight assigned to model ℓ, adjusted for time t. The weights are updated and normalised at each time slot and chosen to give more credit to more recent knowledge. They are chosen from a geometric series $(1,...,(\beta)^\ell))$. After combing the decisions of the models inside every IE, the ensemble will assign a batch with the label of the IE with the highest probability.

4.1.3 Experimental Setup

In order to explore the properties of the proposed framework, we evaluated it on multiple datasets covering various possible scenarios in a multi-camera surveillance system. We conducted our experiments on synthetic as well as real datasets. We run our experiments on a number of CAVIAR video clips [112] including: OneLeave ShopReenter1, Enter ExitCrossingPaths1, OneShopOneWait1, OneStopEnter2, OneStopMoveEnter1, and WalkByShop1front. Due to the presence of different perspectives of the same person, streams are drifting in time.

We employ an automatic tracking approach [135] to track objects in the scene and generate streams of bounding boxes, which define the tracked objects' positions.

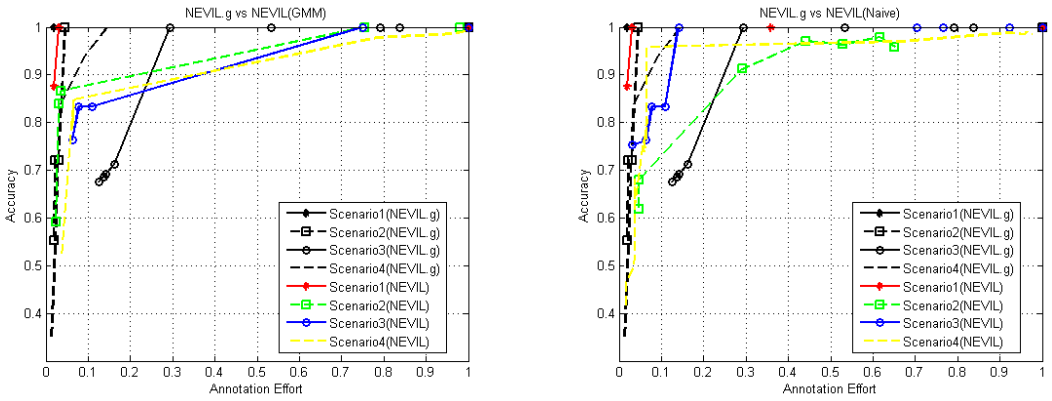
Similar to the experiments in chapter 3, a hierarchical bag-of-visterms method is applied to represent the tracked objects, resulting in a descriptor vector of size 11110 for each frame (refer to [136] for additional details). In order to avoid the curse of dimensionality that system may suffer from, Principle Component Analysis (PCA) is applied to the full set of descriptor features as a pre-processing step. Hence, the number of features in each stream is reduced to 85 dimensions.

4.1.3.1 Random Strategy (Baseline)

The Random strategy [157] labels the incoming batches randomly instead of wisely deciding which batches is more informative. Constrained by budget, batches are sent for annotation.

4.1.3.2 Instantiation of Classifiers

In Section 4.1.2.3, we noted that designing a generative model when we only have positive sample is quite straightforward. However, a challenge may arise since these models output likelihood



(a) (b)

Figure 4.1: Comparison of the performance of NEVIL and NEVIL.gmm on synthetic scenarios

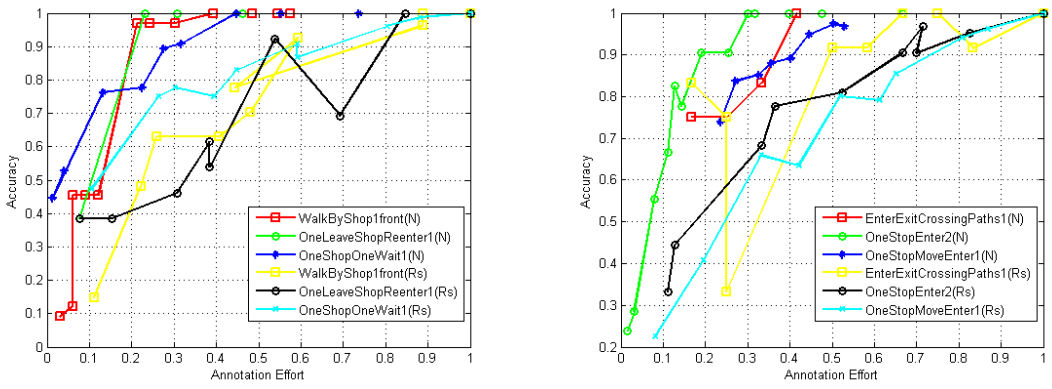
whereas we have made all the decisions based on posterior probability in the theoretical discussion. Nevertheless, assuming equal priors and employing the Bayes' theorem, the likelihood and posterior are consistent.

4.1.4 Results

Firstly, multiple tests were run to determine the optimal batch size for each dataset to be explored. The batch size was varied between 1% to 50% of the size of the longest stream available in each dataset. Experiments were repeated for 50 equally spaced values in that range. The optimal batch size varies and is influenced by the characteristics of the streams present in each dataset. Optimal batch sizes have been observed to range between 30 and 35 for real video streams and between 200 and 300 for synthetic sequences.

Figure 4.1 shows the results of NEVIL.gmm on multiple synthetic scenarios. To the best of our knowledge, there is no approach (except NEVIL) that can classify parallel data streams while interacting with an oracle (there are however clustering approaches which are not applicable in our scenario). Thus we compared this framework with NEVIL in two different settings: GMM-based NEVIL, to be consistent with the results of this version and Naive-based NEVIL, that gave the best results with synthetic data in the previous work. For most of the scenarios, NEVIL.gmm (black plots) outperforms NEVIL and we can get 90% accuracy by only 10% annotation effort.

Figure 4.2 shows that NEVIL.gmm is specially effective when the human collaboration is low. We see that even at big budget Random strategy may fail due to selection of non or less informative batches (see *OneLeave ShopReenter1*). Figure 4.3 presents the results on multiple CAVIAR datasets, where various lengths and number of streams from different classes are present. The most complex scenario is *OneStopMoveEnter1*, with 42 streams from 14 classes. We have compared the results of the proposed method with NEVIL's most successful setting in real video scenarios (SVM-based NEVIL) as well as GMM-based NEVIL. Results show a clear improvement when comparing to GMM-based NEVIL (See Figs. 4.3a, 4.3b). Comparing to SVM-based NEVIL, our method has also improved. For half of the clips (including *OneStopEnter2, WalkBy*



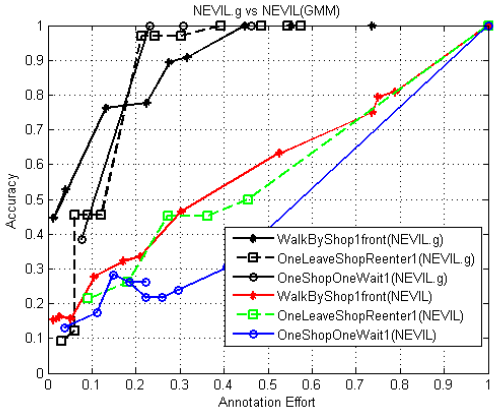
(a) (b)

Figure 4.2: Comparison of the performance of Random Strategy and NEVIL.gmm on real scenarios

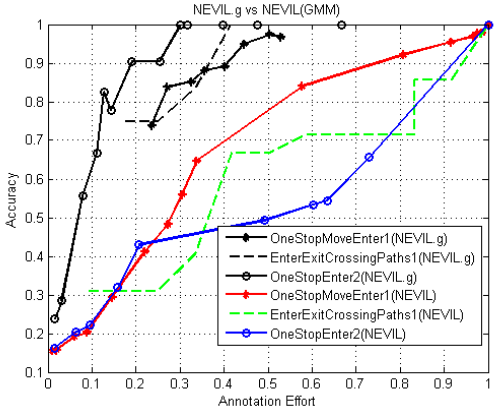
Shop1front,OneLeave ShopReenter1), we obtain over 90% accuracy with a manual labelling of 20% of batches. Although the interaction increase for other scenarios, the value is quite acceptable considering the complexity of the data (we need to annotate 25% of batches to gain 85% correct classification for *OneStopMoveEnter1*).

4.1.5 Discussion

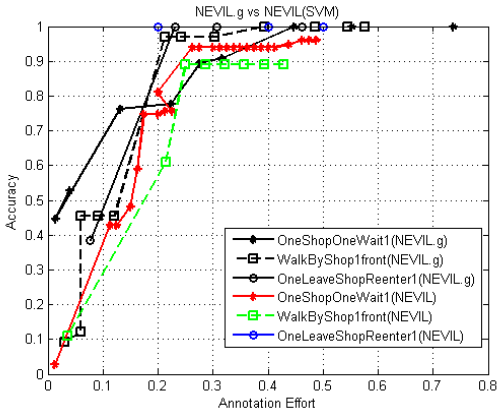
We introduced a class-based ensemble framework for the classification of parallel visual data streams. The framework has shown promising performance with a fairly little human collaboration and can be applied in an on-line process. However, it posses most of the characteristics mentioned in Chapter 1 for our desired learning algorithm, the instability for high-dimensional data, which makes effective novelty detection highly challenging, and the growing complexity are the main concerns.



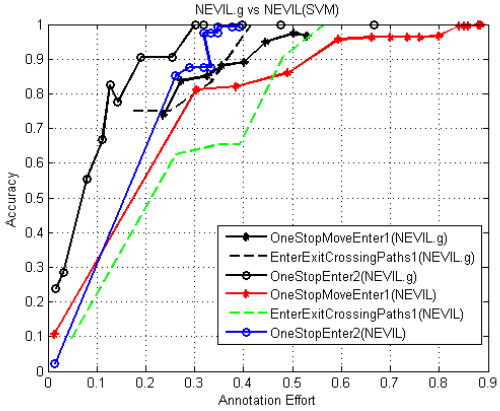
(a)



(b)



(c)



(d)

Figure 4.3: Performance evaluation on multiple CAVIAR clips.

4.2 NEVIL.ubm

Although NEVIL.gmm produces superior performance compared to NEVIL, stability in high-dimensional visual data is still a big issue and the novel class detection is unreliable due to the difficulty of setting a suitable threshold. Here, we address those issues by adopting a UBM-normalized strategy and class-based ensembles. An ensemble of generative models that includes the maximum a-posteriori (MAP) adaptation of the *universal background model (UBM)*. This framework applies a double threshold strategy in order to detect novel classes and unreliable decisions. The decisions are categorized into three groups: novel classes when the existing classes are unable to explain satisfactorily the observed data; unreliable, leading to a request of user input; and reliable when there is strong evidence in favor of one of the existing classes. The adopted batch approach enables to achieve a good balance between the need to have enough data to make reliable decisions and the need to adapt quickly enough to drifts and new concepts in the data streams.

4.2.1 Universal Background Model

Universal background modelling is a common strategy in the field of voice biometrics [111]. It can be easily understood if the problem of biometric verification is interpreted as a basic hypothesis test. Given a biometric sample Y and a claimed ID, S, we define:

H_0: Y belongs to S
H_1: Y does not belong to S

as the null and alternative hypothesis, respectively. The optimal decision is taken by a *likelihood-ratio test*:

$$\mathscr{S}(Y|H_0) = \frac{p(Y|H_0)}{p(Y|H_1)} \begin{cases} \geq \theta & \text{accept } H_0 \\ \leq \theta & \text{accept } H_1 \end{cases} \tag{4.6}$$

where θ is the decision threshold for accepting or rejecting H_0, and $p(Y|H_i), i \in \{0,1\}$ is the likelihood of observing sample Y under hypothesis i. Biometric recognition can, thus, be reduced to the problem of computing the likelihood values $p(Y|H_0)$ and $p(Y|H_1)$. Note that H_0 should characterize the hypothesized individual, while, alternatively, H_1 should be able to model *all the alternatives to the hypothesized individual*.

From such formulation arises the need for a model that successfully covers the space of alternatives to the hypothesized identity. The most common designation in literature for such a model is *universal background model* or *UBM* [115]. Such model must be trained on a large set of data, so as to faithfully cover a representative user space and a significant amount of sources of variability.

4.2.1.1 Hypothesis Modeling

Gaussian Mixture Models (GMM) are typically chosen to model both the UBM, i.e. H_1, and the individual specific models (IDSM), i.e. H_0. Such models are capable of capturing the empirical probability density function (PDF) of a given set of feature vectors, so as to faithfully model their intrinsic statistical properties [114]. The choice of GMM to model feature distributions in biometric data is extensively motivated in many works of related areas. From the most common interpretations, GMMs are seen as capable of representing broad "hidden" classes, reflective of the unique structural arrangements observed in the analysed biometric traits [114]. Besides this assumption, Gaussian mixtures display both the robustness of parametric unimodal Gaussian density estimates, as well as the ability of non-parametric models to fit non-Gaussian data [113]. This duality, alongside the fact that GMM have the noteworthy strength of generating smooth parametric densities, confers such models a strong advantage as generative model of choice.

4.2.1.2 H_1: UBM Parameter Estimation

To train the Universal Background Model a large amount of "impostor" data, i.e. a set composed of data from all the enrolled individuals, is used, so as to cover a wide range of possibilities in the individual search space [126]. The training process of the UBM is simply performed by fitting a k-mixture GMM to the set of feature vectors extracted from all the "impostors".

If we interpret the UBM as an "impostor" model, its "genuine" counterpart can be obtained by adaptation of the UBM's parameters using individual specific data. For each enrolled individual, *ID*, an *individual specific model* (IDSM) is therefore obtained.

4.2.1.3 H_0: MAP Adaptation of the UBM

IDSMs are generated by the *tuning of the UBM parameters* in a maximum *a posteriori* (MAP) sense, using individual specific biometric data. This approach provides a tight coupling between the IDSM and the UBM, resulting in better performance and faster scoring than uncoupled methods [146], as well as a robust and precise parameter estimation, even when only a small amount of data is available [126]. This is indeed one of the main advantages of using UBMs. The determination of appropriate initial values (i.e. seeding) of the parameters of a GMM remains a challenging issue. A poor initialization may result in a weak model, especially when the data volume is small. Since the IDSM is learnt only from each individual data, it is more prone to a poor convergence that the GMM for the UBM, learned from a big pool of individuals. In essence, UBM constitutes a good initialization for the IDSM.

4.2.1.4 Recognition and Decision

After the training step of both the UBM and each IDSM, the typical recognition phase in biometric systems is somewhat trivial. As referred in the previous sections, the identity check is performed through the projection of the new test data, X_{test}, onto both the UBM and either the claimed

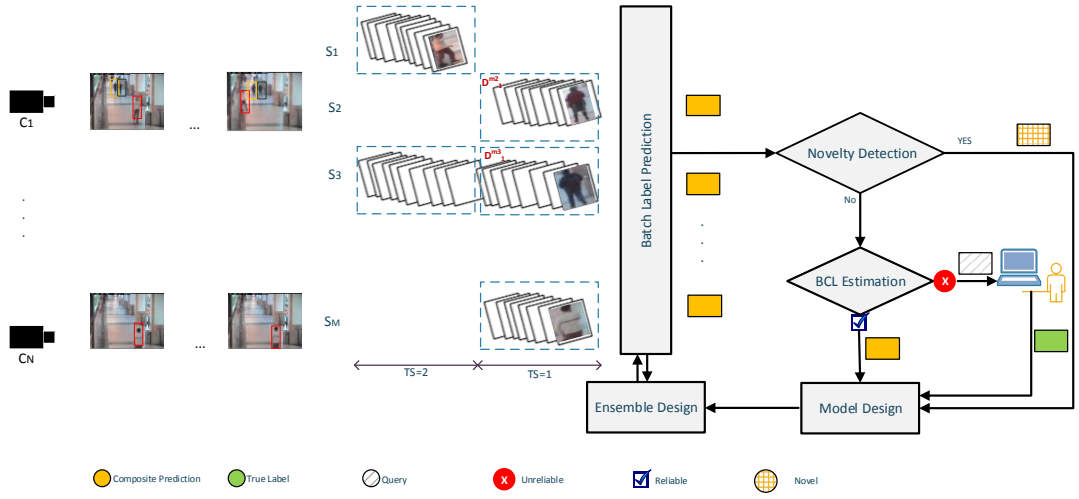


Figure 4.4: NEVIL.ubm High-level Overview

IDSM (in verification mode) or all such models (in identification mode). The recognition score is obtained as the likelihood-ratio. This is a second big advantage of using UBM. The ratio between the IDSM and the UBM probabilities of the observed data is a more robust decision criterion than relying solely on the IDSM probability. This results from the fact that some subjects are more prone to generate high likelihood values than others, i.e. some people have a more "generic" look than others. The use of a likelihood ratio with an universal reference works as a normalization step, mapping the likelihood values according to their global projection. Without such step, finding a global optimal value for the decision threshold, θ, presented in Equation (4.6), would be a far more complex process.

4.2.2 Algorithm Overview

In this section we present the framework named Never Ending Visual Information Learning with UBM (NEVIL.ubm). Algorithm 4 outlines our approach. The framework receives multiple visual streams, generated by a typical tracking algorithm, which analyses sequential video frames and outputs the movement of targets between the frames. Inside each frame the data corresponds to some pre-selected object representation (e.g. bag of words, histogram). Experimentally we will evaluate the stability of NEVIL.ubm with several object representations. Environmental challenges such as varying illumination, lack of contrast, bad positioning of acquisition devices, blurring caused by motion as well as occlusion make data often noisy and/or partially missing. We address these challenges by a batch divisive strategy, as learning from a data batch may reduce the noise and fill the gaps caused by miss-tracking. Initially, the composite model is initialized to yield the same probability to every possible class (uniform prior). When the batches $\mathscr{D}_t^{m_i}$ in time slot t become available, the framework starts computing the scores $\mathscr{S}(\mathscr{D}_t^{m_i}|C_k, H_{t-1})$ for each batch $\mathscr{D}_t^{m_i}$ in the time slot. The scores are obtained from the likelihood ratio test of the batch data obtained

by the individual class model C_k and the UBM. Unrelated objects will sooner or later be assigned the same label or different labels will be assigned to different views of the same object. To avoid propagate such errors, we allow the system to interact wisely with a human, to help it stay on track. Once $\mathscr{S}(\mathscr{D}_t^{m_i}|C_k,H_{t-1})$ is obtained, a batch confidence level (BCL) is estimated; if BCL is high enough (above a predefined threshold), the predicted label

$$\arg\max_{C_k} \mathscr{S}(\mathscr{D}_t^{m_i}|C_k,H_{t-1})$$

is accepted as correct; if the BCL is very low (lower than a pre-determined second threshold), the batch data is assigned to a novel class; otherwise the user is requested to label the data batch. The labelled batches (either automatically or manually) are used to generate new separate models h_t^k (k runs over all the classes available in t), which are then integrated in the composite model, yielding H_t. Four tasks need now to be detailed: a) the batch label prediction (by the composite model); b) novelty detection and batch confidence level estimation c) the individual class model design in current time slot; d) the composite model structure and update.

Algorithm 4 NEVIL.ubm

Input: $\mathscr{D}_t^{m_i}, i = 1,...,M$
$W_0 \leftarrow \frac{1}{k}$
$H_0 \leftarrow W_0$
while $\mathscr{D}_t$ is *True* **do**
  Batch label prediction (Section 4.2.2.1)
  $\mathscr{S}(C_k|\mathscr{D}_t^{m_i},H_{t-1}) \leftarrow (\mathscr{D}_t^{m_i},H_{t-1})$
  Novelty Detection (Section 4.2.2.2)
  $\max_{C_k} \mathscr{S}(C_k|\mathscr{D}_t^{m_i},H_{t-1}) < T \Rightarrow \mathscr{D}_t^{m_i} \subset$ novel class
  Batch Confidence Level Estimation (Section 4.2.2.2)
  $BCL \leftarrow \mathscr{S}(C_k|\mathscr{D}_t^{m_i},H_{t-1})$
  Model design (Section 4.2.2.3)
  $h_t^j \leftarrow \mathscr{D}_t^j, \ \ j = 1,...,k$
  Composite model structure and update (Section 4.2.2.4)
  $ME_t^j \leftarrow h_t^j, \ \ j = 1,...,k$
  $H_t \leftarrow (ME_t^1,...,ME_t^k, \ H_{t-1}, \ W_t)$
end while

4.2.2.1 Batch Label Prediction

A batch $\mathscr{D}_t^{m_i}$ is a temporal sequence of frames $\mathscr{D}_{t,f}^{m_i}$, where f runs over 1 to the batch size B. The composite model, H_{t-1}, can be used to predict directly $p(\mathscr{D}_{t,f}^{m_i}|C_k,H_{t-1})$ but not $p(\mathscr{D}_t^{m_i}|C_k,H_{t-1})$. The individual scores per frame $\mathscr{S}(\mathscr{D}_{t,j}^{m_i}|C_k,H_{t-1})$ can be immediately obtained as $\mathscr{S}(\mathscr{D}_{t,j}^{m_i}|C_k,H_{t-1}) = \frac{p(\mathscr{D}_{t,j}^{m_i}|C_k,H_{t-1})}{p(\mathscr{D}_{t,j}^{m_i}|UBM)}$. The batch label prediction can be analysed as a problem of combining information from multiple (B) classification decisions. Considering that, per frame, the composite model produces approximations to the likelihoods/scores for each class, different combination rules can be considered to build the batch prediction from the individual frame predictions [6, 77]. Assuming

independence between the scores of the individual frames, the score per batch is readily obtained as

$$\mathscr{S}(\mathscr{D}_t^{m_i}|C_k,H_{t-1}) = \sqrt[B]{\prod_{j=1}^{B}\mathscr{S}(\mathscr{D}_{t,j}^{m_i}|C_k,H_{t-1})} \tag{4.7}$$

Some authors have shown that the arithmetic mean outperforms the geometric mean in the presence of strong noise [6, 77]. Thus, as a second option we defined the BCL as:

$$\mathscr{S}(\mathscr{D}_t^{m_i}|C_k,H_{t-1}) = \frac{\sum_{j=1}^{B}\mathscr{S}(\mathscr{D}_{t,j}^{m_i}|C_k,H_{t-1})}{B} \tag{4.8}$$

In our scenario, it is very likely to obtain outlier values for some frames in a batch due to occlusion or miss tracking. The median might be seen as a better indication of central tendency than the arithmetic mean in such cases, since it is less susceptible to the exceptionally large or small values in data. Hence, as a third option we consider estimating the score of a given batch by:

$$\mathscr{S}(\mathscr{D}_t^{m_i}|C_k,H_{t-1}) = \mathscr{M}edian\ \{\mathscr{S}(\mathscr{D}_{t,j}^{m_i}|C_k,H_{t-1}),\ j=1,...,B\} \tag{4.9}$$

Although other robust statistics could be considered from the individual frame scores, experimentally we will only compare the three options.

In the end, NEVIL.ubm assigns each batch to the class maximizing $\mathscr{S}(\mathscr{D}_t^{m_i}|C_k,H_{t-1})$.

4.2.2.2 Novelty Detection and Batch Confidence Level Estimation

In our scenario, the number of classes is unknown beforehand. When a previously unobserved person enters the area of coverage by the camera network, the system should create a new model to represent the novel class. We consider automating this decision. Applying a threshold to detect novel classes is extensively explored in the literature [91].

In our NEVIL.ubm framework, if the scores associated to all observed classes ($\mathscr{S}(C_j|\mathscr{D}_t^{m_i},H_{t-1}), j=1,...,k$) are significantly low (below a predetermined threshold), it is very likely that this class has not observed before and it is considered novel:

$$\max_{C_k}\mathscr{S}(C_k|\mathscr{D}_t^{m_i},H_{t-1}) < T \Rightarrow \text{data belongs to a novel class}$$

Having decided that the batch data belongs to an existing class, one needs to decide if the automatic prediction is reliable and accepted or rather a manual labelling needs to be requested.

As per discussions in Chapter 3, the most commonly used criterion relies on the probability of the most confident class, defining the confidence level as

$$\max_{C_k}\mathscr{S}(C_k|\mathscr{D}_t^{m_i},H_{t-1}) \tag{4.10}$$

Another option is to adopt a margin confidence measure based on the first and second most probable class labels under the model:

$$\mathscr{S}(C^*|\mathscr{D}_t^{m_i}, H_{t-1}) - \mathscr{S}(C_*|\mathscr{D}_t^{m_i}, H_{t-1}), \tag{4.11}$$

where C^* and C_* are the first and second most probable class labels, respectively. Intuitively, batches with large margins are easy, since the classifier has little doubt in differentiating between the two most likely class labels. Batches with small margins are more ambiguous, thus knowing the true label would help the model discriminate more effectively between them [123].

We also proposed a variant of the margin measure, which is based on the *ratio* of the first and second most probable class labels:

$$\mathscr{S}(C^*|\mathscr{D}_t^{m_i}, H_{t-1}) / \mathscr{S}(C_*|\mathscr{D}_t^{m_i}, H_{t-1}), \tag{4.12}$$

4.2.2.3 Model Design

The Universal Background Model is trained offline, before the deployment of the system. UBM is designed from a large pool of streams aimed to be representative of the complete set of potentially observable 'objects'. The training process of the UBM is simply performed by fitting a GMM to the set of feature vectors extracted from the complete pool.

At time slot t, we obtain a new set of batches that are automatically or manually labelled. We assume all the frames belonging to a batch are from the same object and that the M batches in a time slot correspond to $L < M$ labels (some batches can have the same label).

At each time slot, the data from the batches predicted to belong from the same class is used to generate the class model by *tuning of the UBM parameters*, in a maximum *a posteriori* (MAP) sense. This approach provides a tight coupling between the individual model and the UBM, resulting in better performance and faster scoring than uncoupled methods, as well as a robust and precise parameter estimation, even when only a small amount of data is available [126]. The adaptation process consists of two main estimation steps. First, for each component of the UBM, a set of sufficient statistics is computed from a set of M class specific feature vectors, $X = \{\mathbf{x}_1, ..., \mathbf{x}_M\}$ computed from the batch data:

$$n_i = \sum_{m=1}^{M} p(i|\mathbf{x}_m) \tag{4.13}$$

$$E_i(\mathbf{x}) = \frac{1}{n_i} \sum_{m=1}^{M} p(i|\mathbf{x}_m)\mathbf{x}_m \tag{4.14}$$

$$E_i(\mathbf{x}\mathbf{x}^t) = \frac{1}{n_i} \sum_{m=1}^{M} p(i|\mathbf{x}_m)\mathbf{x}_m\mathbf{x}_m^t \tag{4.15}$$

where $p(i|\mathbf{x}_m)$ represents the probabilistic alignment of $\mathbf{x}_m$ into each UBM component. Each UBM component is then adapted using the newly computed sufficient statistics, and considering diagonal covariance matrices. The update process can be formally expressed as:

$$\hat{w}_i = [\alpha_i n_i / M + (1 - \alpha_i) w_i] \, \xi \tag{4.16}$$

$$\hat{\boldsymbol{\mu}}_{\boldsymbol{i}} = \alpha_i E_i(\mathbf{x}) + (1 - \alpha_i) \boldsymbol{\mu}_{\boldsymbol{i}} \tag{4.17}$$

$$\hat{\Sigma}_i = \alpha_i E_i(\mathbf{x}\mathbf{x}^t) + (1 - \alpha_i)(\boldsymbol{\sigma}_{\boldsymbol{i}} \boldsymbol{\sigma}_{\boldsymbol{i}}^t + \boldsymbol{\mu}_{\boldsymbol{i}} \boldsymbol{\mu}_{\boldsymbol{i}}^t) - \hat{\boldsymbol{\mu}}_{\boldsymbol{i}} \hat{\boldsymbol{\mu}}_{\boldsymbol{i}}^{\,t} \tag{4.18}$$

$$\boldsymbol{\sigma}_{\boldsymbol{i}} = \mathrm{diag}(\Sigma_i) \tag{4.19}$$

where $\{w_i, \boldsymbol{\mu}_{\boldsymbol{i}}, \boldsymbol{\sigma}_{\boldsymbol{i}}\}$ are the original UBM parameters and $\{\hat{w}_i, \hat{\boldsymbol{\mu}}_{\boldsymbol{i}}, \hat{\boldsymbol{\sigma}}_{\boldsymbol{i}}\}$ represent their adaptation to the specific class. To assure that $\sum_i w_i = 1$ a weighting parameter ξ is introduced. The α parameter is a data-dependent adaptation coefficient. Formally it can be defined as:

$$\alpha_i = \frac{n_i}{r + n_i} \tag{4.20}$$

The relevance factor r weights the relative importance of the original values and the new sufficient statistics.

4.2.2.4 The Composite Model Structure and Update

Obtaining a meaningful stability-plasticity balance is a key issue in learning from non-stationary environments. The human learning system has addressed this issue by reinforcing existing knowledge that is still relevant, as well as forgetting what may no longer be relevant. The forgetting curve supports the process of forgetting that occurs with the passage of time [119], which is exponential in nature. Inspired by human learning system, a strategic combination of an ensemble of classifiers, that employs dynamically assigned weights, is proposed in [44]. Herein, we applied a time weighted strategy that gives more credit to more recent knowledge. Inspired by the forgetting curve, weights are chosen from the Taylor expansion of an exponential. The IDSM associated to the j^{th} class, h_t^j, is stored in the j^{th} ensemble, the so called Micro ensemble ME_t^j. Contrasting to the classic ensembles, a Micro ensemble includes models that are incrementally trained on incoming batches of a specific class, not all the batches (potentially from multiple classes) in a given timeslot. The composite model H_t is an ensemble of Micro ensembles $ME_t^j, j = 1, ..., K_t$, where K_t is the number of classes observed until time t. Each ME_t^j includes models h_t^j that are trained on incoming batches of the j^{th} class since its appearance until the current time. The outputs of the individual models h_t^j are combined in ME_t^j using a weighted majority voting, where the weights are dynamically updated with respect to the classifiers' time of design. The prediction outputted

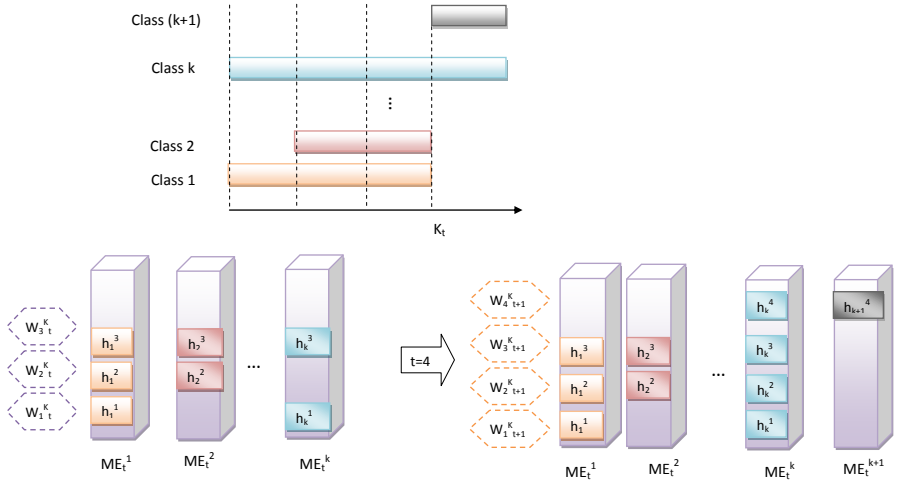


Figure 4.5: An example of composite structure. Once a new class enters the scene (e.g. t=4) , a new micro-ensemble is added to the composite.

by the composite model ME_t^j for a given frame $\mathscr{D}_{t,f}^{m_i}$ is

$$\mathscr{S}(C_k|\mathscr{D}_{t,f}^{m_i}, ME_t^j) = \sum_{\ell=1}^{t} W_\ell^t \mathscr{S}_\ell^j(C_K|\mathscr{D}_{t,f}^{m_i}),$$

where $\mathscr{S}_\ell^j(.)$ is the score outputted by $h_\ell^j(.)$ (the model trained from batches of j^{th} class at TS ℓ), and W_ℓ^t denotes the weight assigned to model h_ℓ^j, adjusted for time t. The weights are chosen from a Taylor expansion of an exponential $(1,...,(\beta)^\ell))$ and are updated and normalised at each time slot to give more credit to more recent knowledge.

Figure 4.5 shows an example of how the composite is updated in a simplified scenario (a class is represented by a single stream). The IDSM associated to each class is trained and stored in the corresponding micro ensemble. For example, classes 1, 2 and k are available in the second timeslot. Hence three IDSM (h_t^1, h_t^2, h_t^k) associated to these classes are stored at ME_2^1, ME_2^2, and ME_2^k, respectively. Once a new class (k+1) appears at $t = 4$ (a new person enters the scene), a new micro ensemble ME_4^{k+1} is built. In order to get a decision on a frame (assign a score to the frame), the outputs of the models are combined using a weighted strategy that gives more credit to the more recent knowledge. Note that these weights are updated at every timeslot.

4.2.3 Experimental Methodology

4.2.3.1 Experimental Design

Besides synthetic datasets, we run our experiments on public indoor (CAVIAR [112], SAVIT-SOFTBIO [15]) and outdoor (PETS) datasets. Seven scenarios of CAVIAR (*OneLeave ShopReenter1, Enter ExitCrossingPaths1, OneShopOneWait1, OneStop Enter2, WalkBy Shop1front*) have been used.

Two views of scenario S2.L1 of PETS2009 have been applied in our experiments. We carry out experiments on *SAVIT* subset of the SAVIT-SOFTBIO, as well. This dataset consists of 11 people moving through a network of 8 cameras. Subjects move in an uncontrolled manner, which provides a highly unconstrained environment reflecting real-world conditions.

We employed an automatic tracking approach [135] to track objects in the scene and generate streams of bounding boxes, which define the tracked objects' positions. As the tracking method fails to perfectly track the targets, a stream often includes frames of distinct objects.

4.2.3.2 RoI Representation

In order to evaluate the stability of the framework, we study the impact of different representations in the performance of NEVIL.ubm. We evaluate three encoding approaches: hard quantization, soft quantization (hierarchical bag-of-visterms), and Fisher method.Classical BoW computes a *spatial histogram (hard quantization)* of visual words constituting the baseline representation.

In order to extract the hard quantized representation, we used a dictionary with 8000 visual words in classic BoW that provides 96000 features for each frame. Following the approach

in [136], a hierarchical bag-of-visterms method is applied to represent the tracked objects, resulting in a descriptor vector of size 11110 for each bounding box (soft quantized representation). In Fisher encoding, visual words are represented by means of a GMM, and the average first and second order differences of image descriptor and the visual words are recorded as global representation. We use a GMM with k=256, resulting in a vector size of 327680 for each bounding box. We used the implementation provided in [23] to extract hard quantized and Fisher Vector features.

To avoid the curse of dimensionality, Principle Component Analysis (PCA) is applied to the full set of features as a pre-processing step. The number of features in each stream is reduced to 85 features for hard quantization and 350 dimensions for both soft quantization and FV.

4.2.3.3 Baseline Methods

We compared our proposed NEVIL.ubm framework with two groups of baseline approaches: 1) unwise methods, in where the query is blindly requested. 2) wise approaches that select queries meticulously.

Unwise Strategy

In such methods (e.g. Random strategy), queries are blindly chosen. The Random strategy [157] labels the incoming batches randomly instead of wisely deciding which batches are more informative. Constrained by budget, batches are sent for annotation.

Wise Methods

To the best of our knowledge, there is no approach (except NEVIL [75] and NEVIL.gmm [74]) that can be used in our learning setting. We stress that the methods in the literature fail to classify uneven parallel streams.

NEVIL [75] trains a classifier (employing discriminative approaches) per time slot. The classifiers are kept in an ensemble and participate in the final decision using a weighted sum strategy. If the decision is not reliable enough, the batch will be sent to an oracle for annotation.

NEVIL.gmm [74] employs a class-based ensemble of GMMs. A computational model is built for individual classes available in a given time slot. Unreliable batches are chosen for manual labeling.

4.2.3.4 Evaluation metrics

ALC, Accuracy, and the amount of annotation effort have been applied to study the characteristics of the framework.

4.2.4 Results

Firstly, multiple tests were run to determine the optimal batch size for each dataset to be explored. The batch size was varied between 1% to 50% of the size of the shortest stream available in each

	ALC			
Datasets	Random	NEVIL	NEVIL.gmm	NEVIL.ubm
Scenario I	0.544	0.976	0.990	0.990
Scenario II	0.532	0.943	0.980	0.986
Scenario III	0.613	0.882	0.886	0.983
Scenario IV	0.523	0.883	0.972	0.973

Table 4.1: Assessment on synthetic datasets.

dataset. Experiments were repeated for 50 equally spaced values in that range. The optimal batch size varies and is influenced by the characteristics of the streams present in each dataset. Optimal batch sizes have been observed to range between 25 and 35 for video streams. In order to explore the properties of the proposed framework, we evaluated it on multiple datasets covering various possible scenarios in a multi-camera surveillance system.

4.2.4.1 Results on Synthetic Data Sets

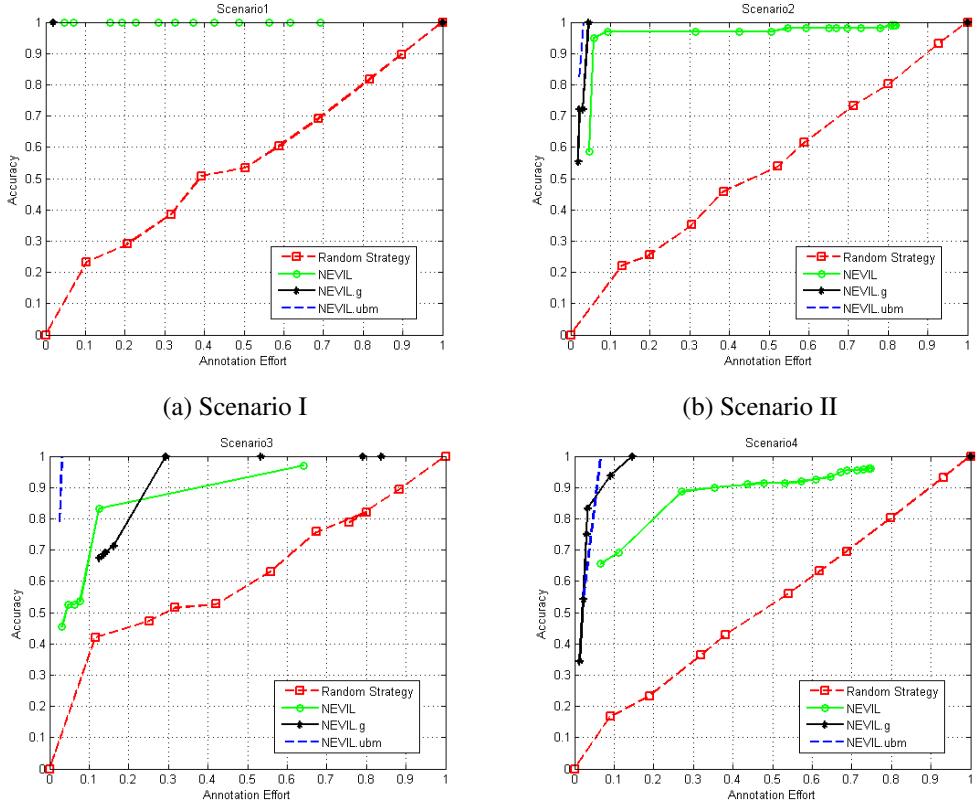
Table 4.1 provides a summary of ALC of baseline approaches as well as NEVIL.ubm on multiple synthetic datasets. We plot the accuracy of a given strategy as a function of annotation effort in Figure 4.6. Results show that NEVIL.ubm outperforms all the other techniques, specially in more complex scenarios: *Scenario III* and *Scenario IV*. All the experiments were repeated 10 times to smooth initialization variability. Results demonstrate that the new framework (NEVIL.ubm) outperforms the baseline methods, providing over 90% accuracy with less than 10% annotation for all the datasets.

4.2.4.2 Results on Real Video Streams

We compared NEVIL.ubm against baseline methods on multiple real video data, where various lengths and number of streams from different classes are present. Results are provided in Table 4.2. We note that the results showed significant improvement in favour of wise strategies in where queries are carefully chosen (Random strategy occupies the lowest place in the table). We observe that NEVIL.gmm and NEVIL.ubm are both significantly better than NEVIL. NEVIL was based on a discriminative learning of the models, being unable to detect novel classes. Therefore, it requires more user input for the same performance. Moreover, the learning of a multiclass classifier at each timeslot using only the subset of objects present in that timeslot is likely to induce false high likelihoods for the more recent classes. NEVIL.ubm has the highest ALC (except for

	Datasets									
Methods	Reenter2	Reenter1	Wait1	front	Path1	Enter2	Enter1	PETS2009	SAVIOT	Mean rank
Random strategy	0.69	0.63	0.59	0.68	0.62	0.66	0.51	0.56	0.57	**4**
NEVIL	0.76	0.90	0.84	0.79	0.74	0.84	0.78	0.68	0.82	**3**
NEVIL.gmm	0.94	0.89	0.90	0.88	0.85	0.91	0.81	0.71	0.88	**2**
NEVIL.ubm	0.93	0.96	0.88	0.93	0.86	0.95	0.87	0.79	0.92	**1**

Table 4.2: Comparison of NEVIL.ubm with baseline methods on real-world datasets.



(a) Scenario I

(b) Scenario II

(c) Scenario III

(d) Scenario IV

Figure 4.6: Performance of baseline methods as well as NEVIL.ubm on synthetic datasets (Accuracy against Annotation effort) The signs denote the results of Random sampling, NEVIL, NEVIL.gmm, and NEVIL.ubm, respectively.

Confidence Measure	Combination Method	Datasets									Mean Rank
		Reenter1	Reenter2	front	Paths1	Enter2	Wait1	Enter1	PETS2009	SAVIOT	
Most Confident Class	Median	**0.96**	0.93	0.93	0.86	0.95	0.88	0.87	0.79	0.88	**3**
	Prod	0.93	**0.97**	0.931	0.85	0.92	0.901	0.89	0.81	0.89	**4**
	Sum	0.96	**0.97**	0.90	**0.87**	0.95	0.90	**0.90**	**0.85**	**0.92**	**1**
Modified Margin	Median	**0.96**	0.91	0.95	0.87	0.93	0.91	0.87	0.71	0.86	**5**
	Prod	0.937	**0.97**	**0.97**	0.84	0.95	0.89	0.87	0.75	0.87	**6**
	Sum	0.962	0.95	0.93	0.85	**0.96**	**0.92**	0.89	0.75	**0.92**	**2**

Table 4.3: Multiple settings of NEVIL.ubm on real-world datasets.

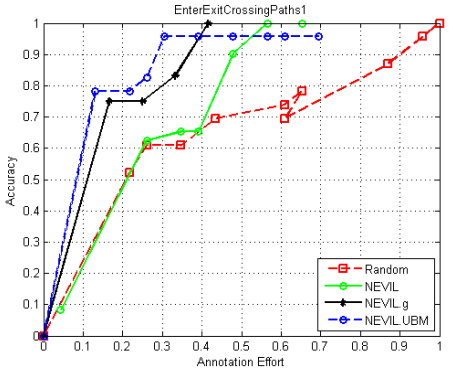
“OneShopOneWait1" and “OneLeaveShopReenter2") and the best mean rank over all the experiments. Figure 4.7 depicts the accuracy of various methods against the amount of queries placed on the operator.

Although there is not a single operating point in the Learning Curve suitable for all the applications, similarly to [32], we chose the point obtained by labelling 20% of batches for a more detailed analysis. Given that budget, we obtain 100% for four scenarios (OneLeaveShopReenter2, OneLeaveShopReenter1, OneStopEnter2, and WalkByShop1front). For more complex scenarios, such as OneStopMoveEnter1 (in where 42 streams from 14 classes are available) 80% of batches are correctly classified, showing a clear improvement over prior approaches. All the results are obtained using the most confident class as batch confidence measure and the median as the combination rule.

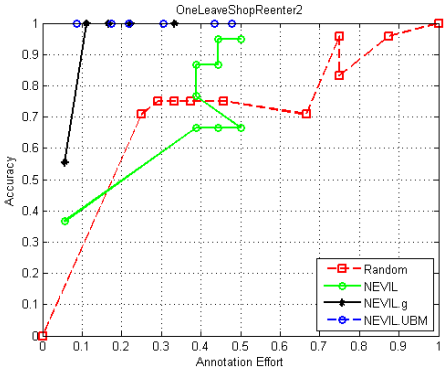
We presented multiple combination rules including sum, product and median and various confidence measure in sections 4.2.2.1 and 4.2.2.2, respectively. Any of the rules and measures can be applied in the framework. Table 4.3 provides a summary of the ALC measure for each setting on all datasets along with the mean of ALC rank averaged over all the experiments. We omit the margin measure results as it has shown results almost equal to the modified margin. The table shows that settings in where sum rule have been applied for combining the information occupy the two of top three spots (first and third). It is not surprising, since sum rule outperformed the product rule when complex data is present [6, 77]. The results indicate that the most confident class as batch confidence measure selects more informative batches than modified margin, as settings employing the former have better mean rank. Based on the average rank, we conclude that the arithmetic mean as combination rule and the most confident as selection criterion represents the optimal design.

Timeline Generation

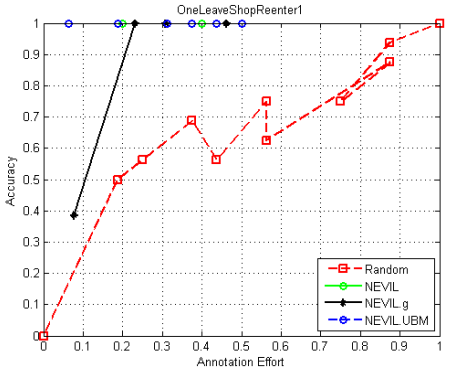
Figure 4.8 shows an example of automatically labelled streams of “OneEnterExitCrossingPath1” and the respective ground truth (Figure 4.8a). The framework assigns labels to the batches. It is desirable to assign the same identifier to all the streams of an individual object, however labels do not carry any semantic information (a name corresponds to a unique number in results). Figure 4.8b shows the output of the framework when 7% of batches are labelled. The framework fails to identify the second class. Figure 4.8c can be considered as a successful case, since all objects are correctly identified. The main difference to the groundtruth is the miss identification of



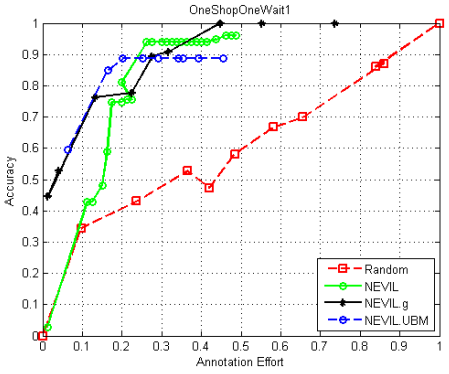
(a) OneLeaveShopReenter2



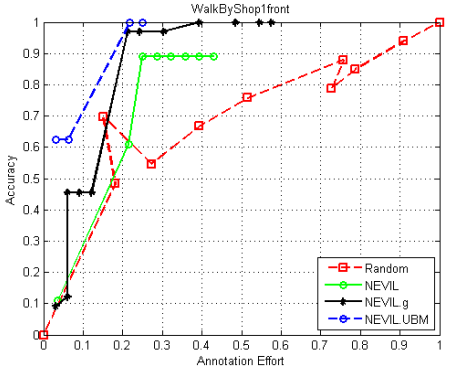
(b) OneLeaveShopReenter1



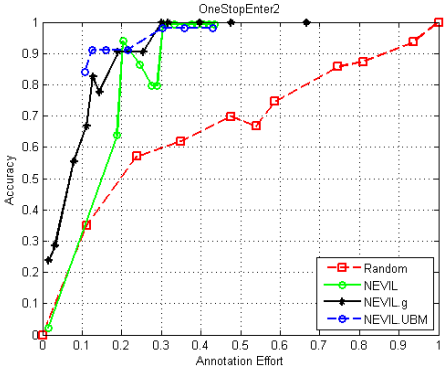
(c) OneShopOneWait1



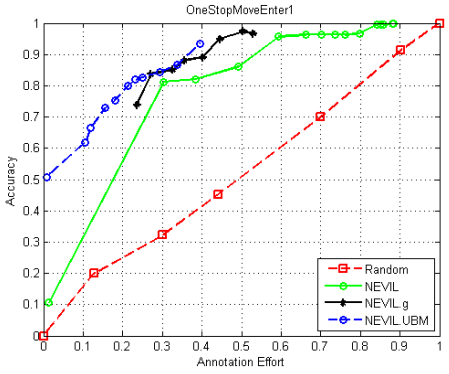
(d) WalkByShop1front



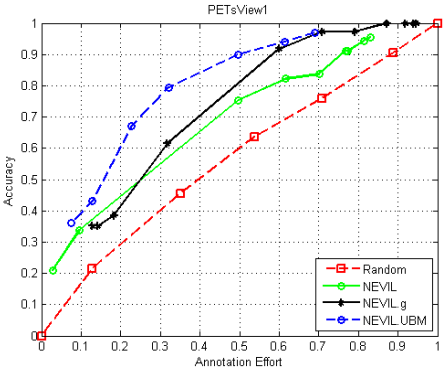
(e) EnterExitCrossingPaths1



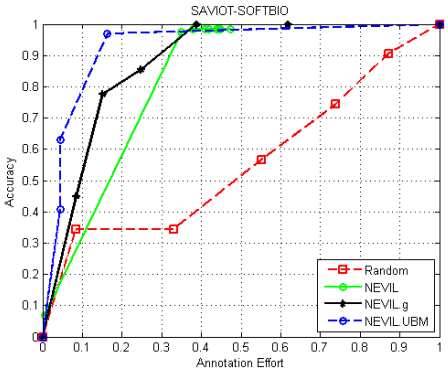
(f) OneStopEnter2



(g) OneStopMoveEnter1

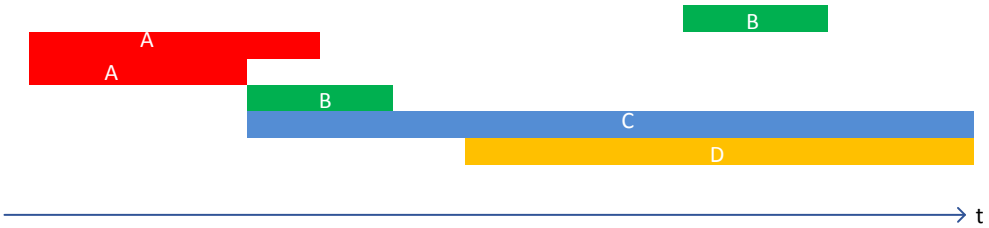


(h) PETS2009

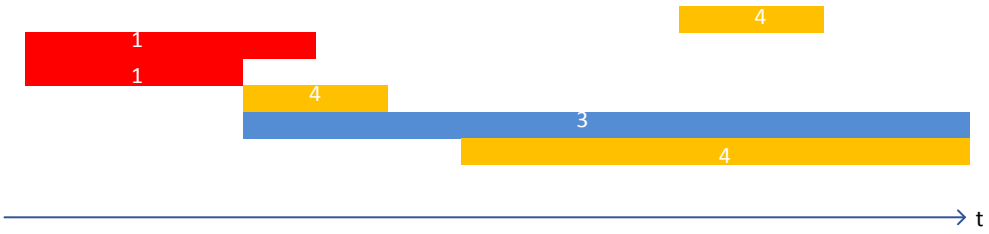


(i) SAVIOT

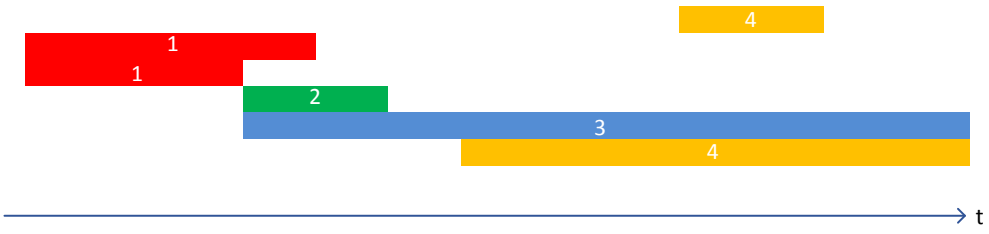
Figure 4.7: Comparison of the performance of NEVIL, NEVIL.gmm, NEVIL.ubm on real-world datasets (Accuracy against Annotation effort). The signs denote the results of Random sampling, NEVIL, NEVIL.gmm, and NEVIL.ubm, respectively.



(a) Groundtruth



(b) Output of the framework using 7% labelling



(c) Output of the framework using 21% labelling

Figure 4.8: Streams of "OneEnterExitCrossingPath1", groundtruth and timeline outputted by the framework using different amount of labelling

a stream. As the second object made a brief appearance in the scene, and he is heavily occluded, the stream experiences an abrupt drift.

4.2.4.3 Stability

In many classical object recognition problems, the representation plays an important role in the performance of the system. Our scenario as a pseudo object classification is not an exception. In here we analyse the impact of the representation in the performance of NEVIL.ubm. We compare the performance reached with the three descriptors introduced in Section 4.2.3.2. Table 4.4 lists FV as the top rank representation, attaining the lowest mean rank. We observe that the performance of NEVIL.ubm does not change much with the representation, presenting a good stability.

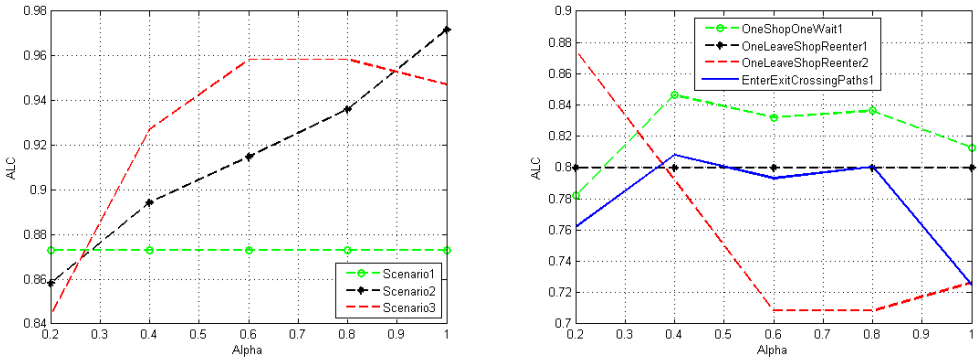
	Hard Quantization	Bag of Vistream	Fisher Kernel
OneLeaveShopReenter2	0.97 (1)	0.95 (2)	0.95 (3)
OneLeaveShopReenter1	0.93 (3)	0.96 (2)	0.97 (1)
OneShopOneWait1	0.91 (2)	0.86 (3)	0.92 (1)
WalkByShop1front	0.79 (3)	0.93 (1)	0.89 (2)
EnterExitCrossingPaths1	0.81 (2)	0.86 (1)	0.76 (3)
OneStopEnter2	0.95 (2)	0.91 (3)	0.97 (1)
OneStopMoveEnter1	0.77 (3)	0.87 (1)	0.85 (2)
PETS2009	0.76 (3)	0.79 (1)	0.79 (2)
SAVIOT	0.79 (3)	0.90 (2)	0.92 (1)
Mean Rank	**3**	**2**	**1**

Table 4.4: The ALC obtained with multiple descriptors. The rank of the descriptors in a given dataset is presented next to the ALC between parentheses.

4.2.4.4 Memory

Decisions made by models inside ensembles are combined in respect to time (ℓ). Models are incrementally forgotten, to give emphasis to models built from more recent data.

To evaluate the impact of the forgetting factor (α), we kept batch size constant letting α vary. Results are plotted in Figure 4.9. We observe that based on the datasets characteristics, exploiting previous models could have different impacts on the final results; for scenarios in where data drifts abruptly and re-reappearance of classes is not present (e.g scenario2, OneShopReenter2), keeping the last model is enough. However, in a real world surveillance system, people may re-enter the scene after a while (which is the case for all our video clips except OneShopReenter2). Furthermore, the appearance of objects may (it is very likely) drift in time, but the drift is not strictly abrupt (which is the case in scenario II, in where data is generated from a completely different distribution). In such scenarios, the framework definitely gets advantage from proper choice of α. Through this proper range the choice of α is not critical (see Figure 4.9b, when $\alpha \in [0.4, 0.8]$).



(a) Synthetic datasets (b) Multiple video clips

Figure 4.9: Effect of forgetting factor (α) in ALC for various synthetic as well as real-world datasets.

4.2.4.5 Time Efficiency

Since NEVIL.ubm was developed in MATLAB without any efficiency concerns, a straightforward assessment of the time efficiency is not adequate. Nevertheless, some comments on the running time are in order. The analysis time grows naturally with the complexity of the dataset; the OneStopMoveEnter1 dataset was therefore the slowest to process. Although the time to process a batch grows with the batch size, since the time spanned by the batch also grows, the overall processing rate is not much affected by the batch size. Finally, ignoring the time to build the UBM model (done before the deployment of the system) the NEVIL.ubm framework was able to process in between real time and twice as fast the video streams, for a framerate of 25fps (running in an Intel Core i7 at 3.2GHz).

4.3 Wrap Up

We propose two frameworks that are intended to learn from uneven parallel streams in non-stationary environments in where both concept drift and concept evolution are available. The framework receives directly the tracked sequences outputted by the tracking system and maintains a global object identity common to all the cameras in the system. Both frameworks share the same characteristics that make them effective for our scenario: 1) interactively query the oracles to stay on track. 2) on-line learning from streams without access to previous data. 3) utilize class-based ensemble to accommodate new knowledge.

As one of the most popular generative approaches, use of Gaussian Mixture Models as the generative approach to train models seems a natural choice. The NEVIL.gmm delivered superior performance compared to NEVIL, however, stability in high-dimensional visual data is still a big issue and the novel class detection is unreliable due to the difficulty of setting a suitable threshold.

NEVIL.ubm adopts a *UBM- normalized* strategy resulting in better and faster decisions as well as a robust and precise classifiers parameter estimation, even when only a small amount of data is available. This framework has shown promising performance with a fairly little human

collaboration. However, growing complexity in terms of runtime and memory is still the main concern for long-term observation.

Chapter 5

Batch Representation

Video surveillance applications, such as activity recognition, are increasingly making use of multiple sensors and modalities. The fusion of multiple diverse sources of information is expected to benefit the system for the recognition of objects, persons, activities and events captured in an array of cameras.

Environmental challenges such as the variation in appearance of individuals due to changes in illumination, contrast, positioning of acquisition devices, motion blur as well as occlusion lead to noisy and/or partial RoI captures. These challenges have previously been addressed by a batch divisive strategy, that views a batch of RoI as a unique element to classify, since learning from these batches may reduce noise and fill the gaps caused by dropped-tracks. A batch includes a fixed number of consecutive RoIs (sources of information) of a given stream and a single label is assigned to the batch of same person in time. Each batch of RoI can be learnt using either a discriminative or a generative classifier, and the pool(s) of classifiers generated in one or more FoVs can be combined into an ensemble of classifiers. Fusion of multiple sources into an ensemble have been addressed by three main approaches in the literature: early, mid-level, and late [38]. Early fusion combines the information in the first possible level (so called signal level fusion in image processing), whereas late fusion combines the information as late as possible (decision level fusion) [1,66]. Mid-level fusion is an interesting compromise that combines the information in an intermediate abstraction level [121].

Score-level fusion as the most popular way of fusion has been employed in previous frameworks (e.g. Section 4.2.2.1). A quantitative similarity measure disseminates valuable information about the input, and yet it is still easy to process compared to sensor-level or feature-level data. However the score space is subject to considerable flexibilities, e.g. different normalization methods may lead to different decision boundaries. Furthermore, small number of scores in a batch might easily overfit the data [134]. On the other hand, feature-level fusion schemes derive the most abstract form of original multiple feature set by eliminating redundant information. The advantages of this scheme are the use of only one learning stage to combine the information (instead of running individual learning stage for every single feature set) for rapid decisions.

In this chapter, two feature-level abstraction schemes that represent the entire batch with a

single descriptor are proposed. These descriptors are obtained by combining features of individual frames in different ways. To the best of our knowledge, this work is the first attempt to explore spatial-temporal fusion schemes for RoI batches captured from video streams generated in a multi-camera surveillance scenario. Note, this study will be conducted under the NEVIL.ubm (version of the) framework.

5.1 Spatial-Temporal Fusion Schemes Over Frames

Early fusion has been applied to define whether the audio signal is consistent with the speaker video file [49]. Pixel-level fusion has shown promising performance in video-based biometric recognition [30] as well as multiple object tracking [33]. Some authors demonstrate [68, 80] demonstrated the effectiveness of the decision level fusion strategies on object tracking, video segmentation, and video event detection. Feature-level fusion has gained much importance over the past few years, and various approaches have been introduced in the literature [1, 26, 125]. Most approaches combined the information of multiple modalities (sensors), while some methods used the complementary descriptors. The former requires multiple sensors (visible light cameras combined with depth or infra-red camera), and the latter adds more complexity to the system specially in an online application. To the best of our knowledge, the employment of feature-level techniques over frames in a Person Re-ID scenario has not been addressed before.

Fusion schemes have been successfully used in large-scale recognition systems to address multiple issues confronting these systems such as accuracy, practicality, and efficiency. Inspired by the rationale behind such systems, two fusion schemes to combine the information in a Person Re-ID system are proposed. Each frame can be considered as an independent source of information and combining such information in different levels could be beneficial for a Person Re-IDsystem. The batch score ($\mathscr{S}(\upsilon_t^{m_i}|C_k,H_{t-1})$) can be obtained in two ways: either by combining the scores of individual RoIs in a batch (score-level fusion), or by combining the patterns of M RoIs in a batch (feature-level fusion).

5.1.1 Feature-Level Fusion

Finding a joint representation for a group of frames is a challenging problem in visual applications. There is a considerable body of research works that addressed this problem by choosing a key frame, which represents the entire batch. As the quality of the batch representation relies heavily on the representative sample and an inappropriate choice may lead to unreliable results, such methods seem impractical for challenging environments. This is the main rationale behind approaches exploiting fusion schemes. In this paper, two feature-level fusion that aggregate descriptors of all the frames in a given batch are proposed. Let $\upsilon_{t,f}^{m_t}$ be the descriptor of $f-th$ frame in a batch, the average histogram that combines the information of entire batch in a single

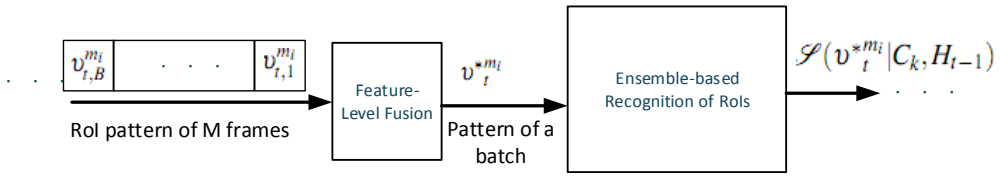


Figure 5.1: Block diagram of feature-level fusion (performed before micro-ensembles recognition)

histogram defined by:

$$\upsilon_t^{*m_i}(b) = \frac{1}{B}\sum_{f=1}^{B} \upsilon_{t,f}^{m_i}(b) \;\; where \;\; b = 1,...,M \tag{5.1}$$

Where M is the number of histogram bins.

In our scenario, it is very likely to obtain outlier values for some frames in a batch due to occlusion or miss tracking. The median might be seen as a better indication of central tendency than the arithmetic mean in such cases, since it is less susceptible to the exceptionally large or small values in data. Hence, as an alternative option we consider estimating the descriptor of a given batch by:

$$\upsilon_t^{*m_i}(b) = \mathscr{M}edian_{f=1,...,B} \;\; \upsilon_{t,f}^{m_i}(b) \;\; where \;\; b = 1,...,M \tag{5.2}$$

Given the single representation, a score $\mathscr{S}(\upsilon_t^{m_i}|C_k,H_{t-1})$ is calculated for the batch.

5.1.2 Score-Level Fusion

The composite model, H_{t-1}, can be used to predict directly $p(\upsilon_{t,f}^{m_i}|C_k,H_{t-1})$ but not $p(\upsilon_t^{m_i}|C_k,H_{t-1})$. The individual scores per frame $\mathscr{S}(\upsilon_{t,j}^{m_i}|C_k,H_{t-1})$ can then be immediately obtained as $\mathscr{S}(\upsilon_{t,j}^{m_i}|C_k,H_{t-1}) = \frac{p(\upsilon_{t,j}^{m_i}|C_k,H_{t-1})}{p(\upsilon_{t,j}^{m_i}|UBM)}$. The batch label prediction can be analysed as a problem of combining information from multiple (B) classification decisions. Considering that, per frame, the composite model produces approximations to the likelihoods/scores for each class, different combination rules can be considered to build the batch prediction from the individual frame predictions. Applying arith-

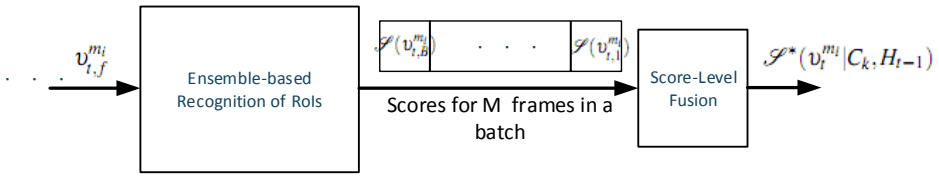


Figure 5.2: Block diagram of score-level fusion (performed after micro-ensembles recognition)

Confidence Measure	Combination Rule	Reenter1	Reenter2	front	Paths1	Enter2	Wait1	Enter1	PETS09
MC	Median	0.96(1)	0.96(2)	0.91(4)	0.87(2)	0.96(1)	0.90(4)	0.90(1)	0.85(2)
	Mean	0.94(3)	**0.97(1)**	**0.96(1)**	**0.88(1)**	**0.96(1)**	**0.91(3)**	**0.90(1)**	**0.86(1)**
MM	Median	0.95(2)	0.94(3)	0.93(3)	0.86(3)	0.96(1)	0.93(1)	0.88(2)	0.79(3)
	Mean	0.96(1)	0.96(2)	0.95(2)	0.87(2)	0.96(1)	0.92(2)	0.90(1)	0.73(4)

Table 5.1: ALC of fusion at feature-level on videos. The rank of each setting in a given dataset is presented next to the ALC between parentheses. Highlighted row indicates the optimal design. Values in bold indicate better performance than score-level fusion for optimal setting.

metic mean, the score per batch is obtained as:

$$\mathscr{S}^*(\upsilon_t^{m_i}|C_k,H_{t-1}) = \frac{\sum_{j=1}^{B}\mathscr{S}(\upsilon_{t,j}^{m_i}|C_k,H_{t-1})}{B} \tag{5.3}$$

As an alternative choice, the median of the scores were also evaluated, since it may be more robust to the outliers. The batch score is defined by:

$$\mathscr{S}^*(\upsilon_t^{m_i}|C_k,H_{t-1}) = \mathscr{M}edian\mathscr{S}(\upsilon_{t,j}^{m_i}|C_k,H_{t-1}) \tag{5.4}$$

Although other robust statistics could be considered from the individual frame scores, experimentally we will only compare the two options. In the end, NEVIL.ubm assigns each batch to the class maximizing $\mathscr{S}(\upsilon_t^{m_i}|C_k,H_{t-1})$.

5.2 Experimental Setup

As before, the adopted representation is a hierarchical bag-of-visterms method is applied to represent the tracked objects, resulting in a descriptor vector of size 11110 for each frame (refer to [136] for more information). In order to avoid the curse of dimensionality that system may suffer from, PCA is applied to the full set of descriptor features as a pre-processing step. Hence, the number of features in each stream is reduced to 85.

5.3 Results

Table 5.1 shows the ALC performance of the proposed fusion techniques using all datasets along with the mean of ALC rank averaged over all the experiments (the std of the results is always

		Datasets							
Confidence Measure	Combination Rule	Reenter1	Reenter2	front	Paths1	Enter2	Wait1	Enter1	PETS09
MC	Median	0.96(1)	0.93(3)	0.93(2)	0.86(2)	0.95(2)	0.88(4)	0.87(3)	0.79(2)
	Mean	**0.96(1)**	**0.97(1)**	0.90(3)	0.87(1)	0.95(2)	0.90(3)	0.90(1)	0.85(1)
MM	Median	0.96(1)	0.91(4)	0.95(1)	0.87(1)	0.93(3)	0.91(2)	0.87(3)	0.71(4)
	Mean	0.96(1)	0.95(2)	0.93(2)	0.85(3)	0.96(1)	0.92(1)	0.89(2)	0.75(3)

Table 5.2: ALC of fusion at score-level on videos. The rank of each setting in a given dataset is presented next to the ALC between parentheses. Highlighted row indicates the optimal design. Values in bold indicate better performance than score-level fusion for optimal setting.

below ± 0.01). The table shows that settings in where sum rule have been applied for combining the information occupy the two top spots for both feature-level and score-level fusion. The results indicate that the most confident class as batch confidence measure selects more informative batches than modified margin, as settings employing the former have better mean rank. Based on the average rank, we conclude that the arithmetic mean as fusion rule and the most confident as selection criterion presents the optimal design. Comparing the ALC of identical designs of two fusion schemes (highlighted rows in tables 5.1 and 5.2) for every dataset, we observe that for 6 out of 8 datasets feature-level fusion attains better performance (higher ALC) than score-level fusion.

Figure 5.3 presents the results of optimal design (arithmetic mean as fusion rule and the most confident as selection criteria) for two fusion levels on all video clips. Since ALC measures the average performance over various budget levels, it does not give detailed information for every single budget level. We chose the point obtained by labelling 20% of batches for a more detailed analysis. Given that budget while employing mid-level fusion, we obtain 100% accuracy for four scenarios (OneLeaveShopReenter2, OneLeaveShopReenter1, OneStopEnter2, and WalkByShop1front). For more complex scenarios, such as OneStopMoveEnter1 (in where 42 streams from 14 classes are available) 88% of batches are correctly classified, showing an improvement over score-level fusion results (80% accuracy). The results indicate the better performance of feature-level over score-level fusion.

Complexity Improving the accuracy is not the only advantage of feature-level fusion. In real-time learning, when massive amount of information is available, efficiency is equally important. In contrary to score-level fusion, where an independent recognition process is applied to every single RoI (of M RoIs in a batch) and then the results are mathematically combined, feature-level fusion employs a single learning stage on the joint representation of a batch of M frames. Thus, the time and complexity of the framework decrease dramatically. Since the framework was developed in MATLAB without any efficiency concerns, a straightforward assessment of the time efficiency is not adequate. Nevertheless our experiments shows that combining the information at feature-level is able to process the streams almost twice as fast as score-level fusion, for a framerate of 25fps (running in an Intel Core i7 at 3.2GHz).

5.4 Discussion

Most state of art approaches for a group of frames representation under specific capture conditions (i.e. uncluttered background), thus they may fail to output an effective representation in a less-controlled situation. Gait-based algorithms represent video shots with gait features. Since extracting distinctive features usually requires a controlled environment (i.e. uncluttered background, consistent silhouette extraction), these methods are not applicable in our scenario [67, 83, 103, 104, 143]. Tensor-based representations deliver a successful performance for some applications (e.g. content-based video retrieval, shot boundary detection) [147], however they require

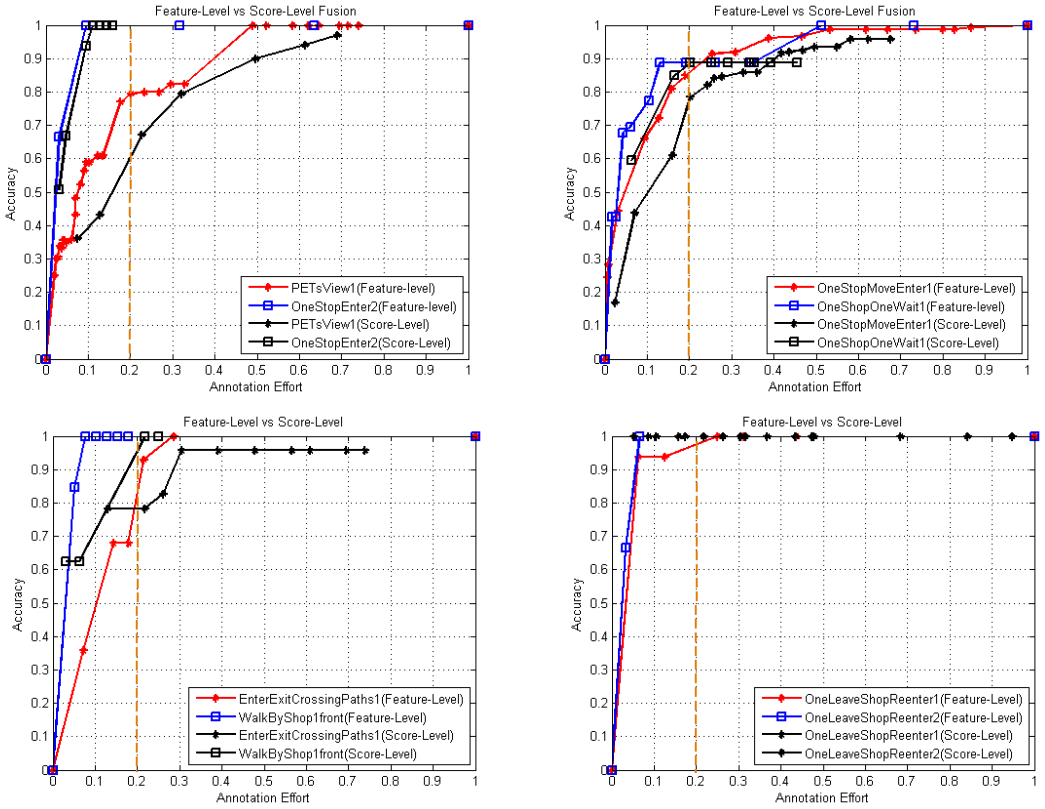


Figure 5.3: ALC vs annotation effort for feature-level with score-level fusion on the various videos. – – – highlights 20% budget.

fixed-size batches to output a comparable representation. In our scenario, the number of RoIs inside a batch is a variable parameter under the limit of B. Hence, the tensor-based methods are not directly applicable in our scenario.

In this chapter, two spatio-temporal fusion strategies, which do not necessary need a controlled environment, to combine the patterns of RoIs in various streams are presented. We experimentally investigated the impact of feature-level and score-level fusion on the performance of the Person Re-ID system. Experiments indicate the potential of feature-level fusion for on-line applications, as they attained the best performance with much lower time complexity.

Chapter 6

Context-sensitive Representation

The Longman dictionary defines context as "the situation, events, or information that are related to something and that help you to understand it". Using context has a history in computer vision [54, 56, 60, 149], however using context information for person Re-ID is a less explored area. Some research works [131] have shown that better quality of data leads to more accurate models. Thus it is crucial to select "good" frames from which to learn a model for long-term tracking. They proposed a supervised criterion that automatically selects the most truth-worthiest frames, however such approach requires large sets of negative examples to train an effective model. In this chapter, we put forward a batch representation that exploits the context information to output an effective context-sensitive representation.

6.1 Representation

In chapter 5 a solo representation per batch is presented. Inspired by [131], it is expected using context information may improve the quality of the descriptor. We propose an unsupervised quality criterion, $Q(RoI_f)$, that automatically gives more credit to more trustworthy RoIs. When two persons are very close, the detected bounding boxes for either of them may contain many pixels of the other, or when objects enter or leave the scene, they are likely only partially visible. $Q(RoI_f)$, which consists of two *unsupervised criteria*, Disjuncture Measure ($Q_1(RoI_f)$) and Peripheral Measure ($Q_2(RoI_f)$), tries to tackle these situations.

The weight assigned to the f_{th} RoI of the m_{th} stream at timeslot t, $Q_t^m(RoI_f)$, is formulated as:

$$Q(RoI_f) = Q_1(RoI_f)Q_2(RoI_f) \tag{6.1}$$

where for the reason of clarity all the subscripts and superscripts are omitted.

Let $\upsilon_{t,f}^m$ be the descriptor vector of the f_{th} RoI in a batch belonging to m_{th} stream. The solo descriptor of the batch, υ_t^{*m}, that combines the information of entire batch with respect to the

quality measure in a single vector defined by:

$$\upsilon^{*m}_{t} = \frac{\sum_{f=1}^{B} Q^{m}_{t}(RoI_f)\upsilon^{m}_{t,f}}{\sum Q^{m}_{t}(RoI_f)} \tag{6.2}$$

Note, that Eq. 6.2 generalizes the formula 5.1 in chapter 5.

6.2 Context Quality Measures

Disjuncture Measure (DM) Suppose that, for a certain video stream, there are multiple RoIs in a given frame at t. The DM ($Q_1(RoI_f)$) is defined by:

$$Q_1(RoI_f) = \frac{\mathscr{A}(RoI_f) - \mathscr{A}(RoI_f \bigcap (\bigcup RoI_{j,j\neq f}))}{\mathscr{A}(RoI_f)} \tag{6.3}$$

where $\mathscr{A}$ is the area of a region. Q_1 is expressed on a range from 0 to 1. $Q_1 = 1$ indicates that the RoI_f is visible and has no intersection with other RoIs in a given frame.

Peripheral Measure When a person enters the field of view of a given camera, quality is likely to be low, due to the inherent noise in the of the borders of the image (out of focus, radial distortion, etc.) and due to the "warm-up" process of the tracker when the tracking is initialized. We assume that quality tends to be higher when the center of the RoI is close to the center of the frame. We define the Peripheral Measure, Q_2, as:

$$Q_2(RoI_f) = 1 - (max(\frac{x - x_c}{a}, \frac{y - y_c}{b}))^{\gamma} \tag{6.4}$$

where, (x, y) and (x_c, y_c) are the center of RoI and frame, respectively. γ allows to control the steepness of the curve.

6.3 Experimental Setup

RoI Representation Our reference image descriptor is an improved version of FV, since the FV was found to serve as the most effective encoding technique for pooling approaches in recent studies [25]. Given an image (RoI), the IFV υ is obtained by extracting a dense collection of patches and corresponding local image features (herein, SIFT [89]) from the image at multiple scales. Each descriptor ($x_i \in R^D$) is then soft-quantized using a Gaussian Mixture Model with K components. FV captures the average first and second order differences between the image descriptors and the centres of the GMM, leading to 2KD-dimensional image descriptor. We used GMM with K = 256, resulting in a vector size of 327680 for each bounding box. We used the implementation provided in [23] to extract Fisher Vector features. To avoid the curse of dimensionality, Principle Component Analysis (PCA) is applied to the full set of features as a pre-processing step. The number of features in each stream is reduced to 200 dimensions.

Batch Representation Most approaches assume that RoIs have the same quality (obviously is not true for real-world scenarios), while in this work a RoI quality criterion is also considered. Once the RoI wise features are extracted, a weighted average over time, which gives more credit to more trustworthy RoIs, is adopted to produce a batch wise representation.

6.4 Do more trustworthy RoIs help the performance?

The main objective of these experiments is to analyse how well context quality weighting incorporate into performance of learning framework. First, we evaluate the impact of the measures individually as well as combination. Then, we explore whether more trustworthy RoIs lead to learn more effective models.

6.4.1 Impact of Exploiting Q_1 Measure

Figure 6.1 illustrates the impact of Q_1 measure on the performance of the framework. The plots define the ALC as a function of annotation. In datasets, i.e PETS, EnterExitCrossingPath1, and OneShopOneWait1 where the people cross, employing intersection-weighted feature vector improves the results.

6.4.2 Impact of Exploiting Q_2 Measure

Fig. 6.2 shows a sample of miss-tracking RoIs in "EnterExitCrossingPath1" dataset. The a, b, and γ are defined $\frac{RoI\ width}{2}$, $\frac{RoI\ height}{2}$, and 4, respectively. Using this technique has increased the area under learning curve for "EnterExitCrossingPath1" from 0.84 to 0.85.

6.4.3 Impact of Exploiting Q Measure

We present the results of exploiting the quality measure to obtain a single descriptor for individual batches at the second row of table 6.1. Adding context quality measures into the representation term of batches improve the performance for all three datasets.

As the next step, *we study whether using quality measures to weight the decisions of classifier improve performance?* Given a set of RoIs ($RoI_f \quad f = 1,...,B$) used to learning a classifier, the weight assigned to the decision of this model ($\mathscr{W}_h$) is obtained by:

$$\mathscr{W}_h = \frac{\sum_{f=1}^{B} Q(RoI_f)}{B} \tag{6.5}$$

The decision made by models inside ensembles are combined using this quality-weighted strategy. Results (third row of Table 6.1) shows an improvement comparing to the first row (where no contextual information have been considered).

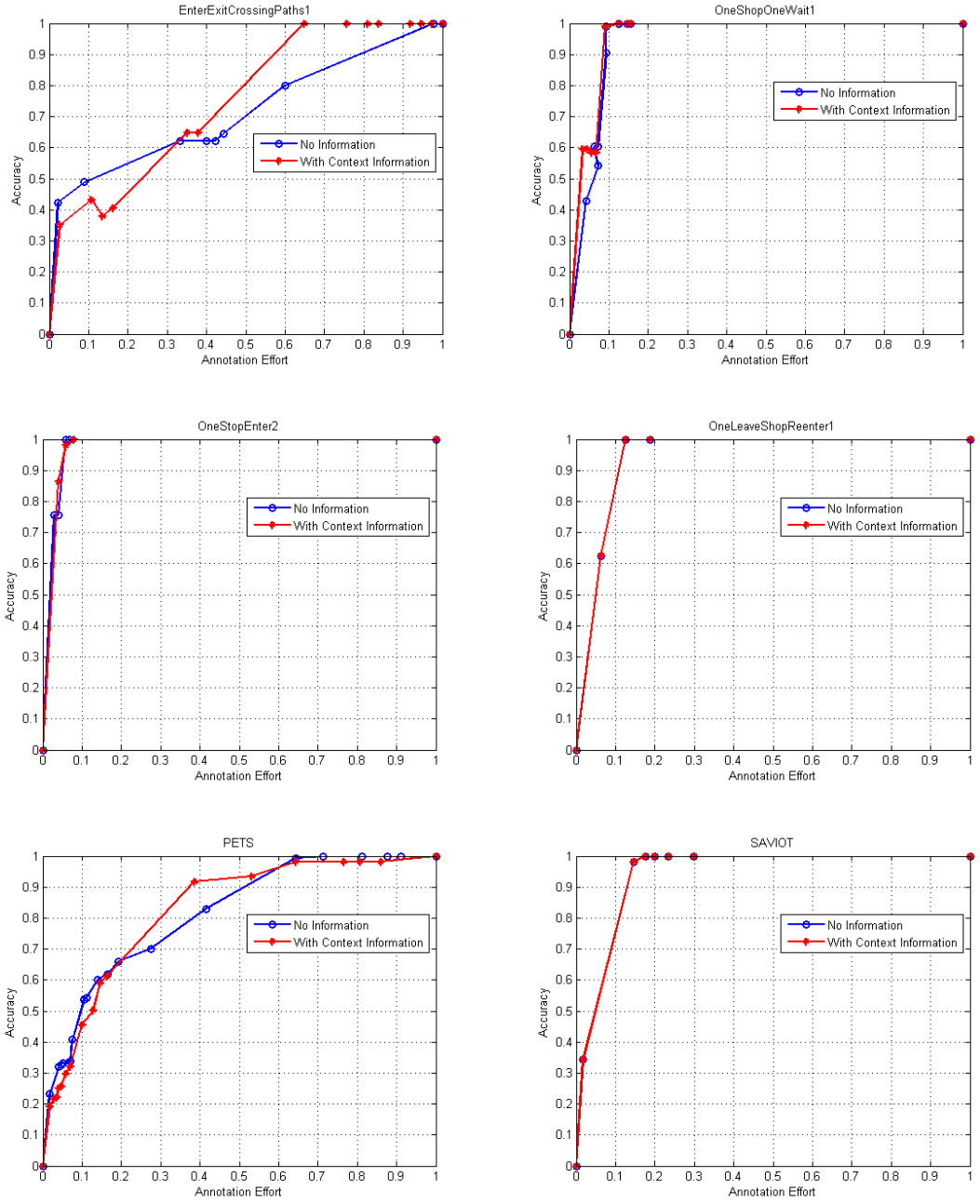


Figure 6.1: Impact of exploiting Q_1 to give more credit to cleaner data on various video dataset. The Accuracy is presented as a function of Annotation effort. The signs $\multimap$ denote the results of ignoring and considering context information, respectively.

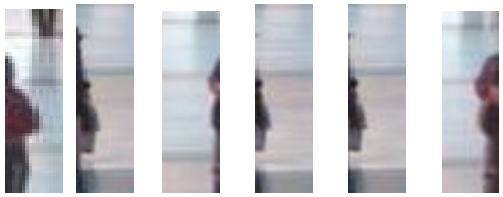


Figure 6.2: sample of partial tracking

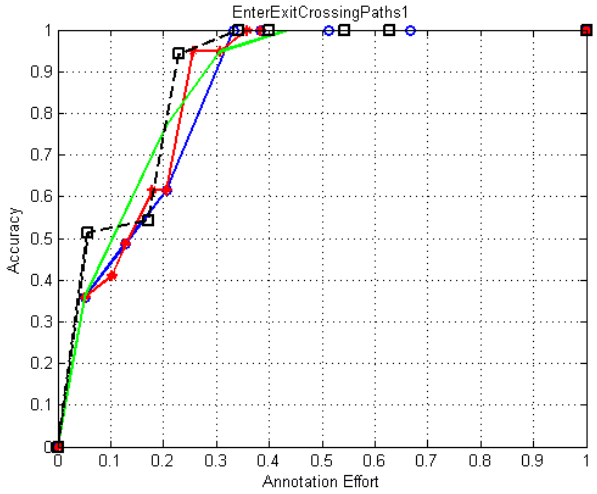
Methods	Datasets			Mean rank
	EnterExitCrossingPath1	OneShopOneWait1	PETS	
NO context information [73]	0.842 (4)	0.923 (3)	0.833 (4)	**4**
Context sensitive batch representation	0.864 (3)	0.957 (1)	0.851 (2)	**2**
Context sensitive learner decisions	0.873 (2)	0.935 (2)	0.844 (3)	**3**
Fusion at feature and decision level	0.885 (1)	0.957 (1)	0.860 (1)	**1**

Table 6.1: Results of exploiting context information at data-level(second row) and model-level(third) on video datasets

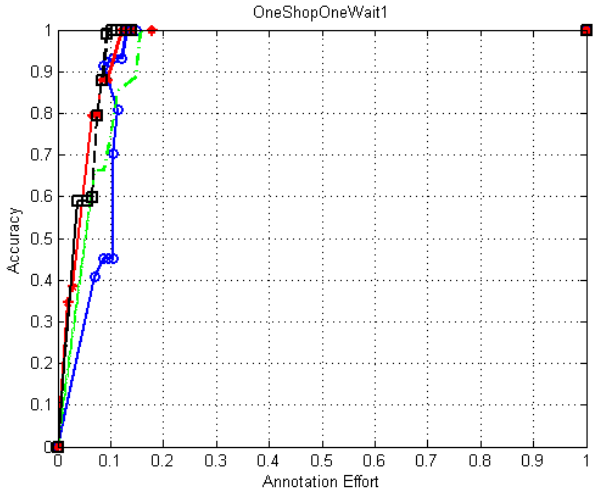
More trustworthy RoIs leads to more effective models as well as more accurate predictions. Here, we investigate the impact of exploiting both context-sensitive representation and quality-weighted decisions under the NEVIL.ubm (version of the framework). Table 6.1 provides a summary of the ALC measure for each setting on three datasets along with the mean of ALC rank averaged over all the experiments. The table shows that settings in where quality measures have been applied for fusing context-sensitive representation and quality-based decisions occupies the top spot, showing a clear improvement over experiments inwhere no context information is considered. Context-sensitive descriptors provide the second successful strategies.

6.5 Discussion

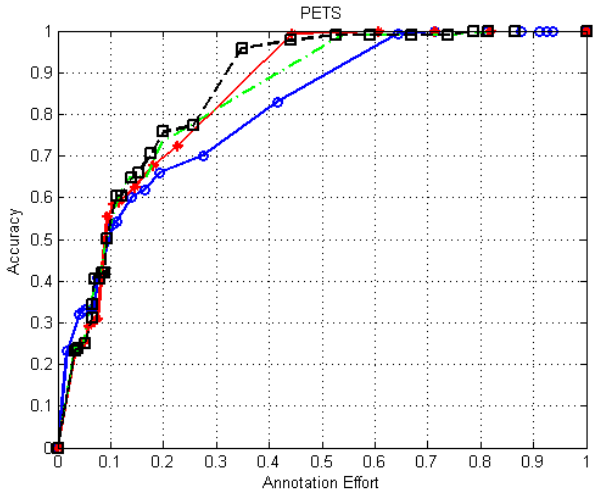
The framework receives directly the tracked sequences outputted by the tracking system and maintains a global object identity common to all the cameras in the system. We proposed a criterion to find more trustworthy RoIs. We further combine the information of RoIs to output a single representation vector per batch. The results indicate that the proposed criterion has the potential to properly rank the RoIs and it is worthwhile to give more credit to better RoIs, resulting in higher performance.



(a) EnterExitCrossingPaths1



(b) OneShopOneWait1



(c) PETS

Figure 6.3: Comparison of the performance of feature-level and model-level intelligent ensemble development on real-world datasets (ALC against updating ratio). The signs -•- -•- denote the results of feature-level and model-level, respectively.

Chapter 7

INEVIL

A classic approach to track changes is to train new classifier(s) as new data arrives and to keep all the classifiers [41]. Accumulating large number of classifiers imposes serious costs (i.e. acute storage space an long prediction time) to the system. Although the costs seems negligible with relatively simple research datasets, they may become highly critical for complex real-word data. In fact, these approaches can easily generate thousands of classifiers under a time-evolving environments. Additionally, it is not always true that the bigger ensemble, the better it is [151].

In previous versions of the framework, we tried to address this problem using a time-weighting strategy, in which decisions made by models inside ensembles are combined in respect to time. However, by giving higher weights to the decision made by more recent models, the older ones are forgotten in time, still a substantial number of models are kept in the framework.

In this chapter, the Intelligent Never Ending Visual Information Learning (INEVIL) framework is presented. INEVIL is specifically designed for long-term learning from non-stationary environments in which no labelled data is available but the learning algorithm is able to interactively query teachers to label meticulously chosen observations.

7.1 Learning Framework

The system receives multiple visual streams, generated by a typical tracking algorithm, which analyses sequential video frames and outputs RoIs over time. A batch is a temporal sequence of RoIs, RoI_f, where f runs over 1 to the batch size B. A high-level sketch of the proposed method is illustrated in Figure 7.1. Five tasks need to be detailed: a) the batch representation; b) batch score estimation; c) batch confidence level estimation; d) batch label assignment; e) Intelligent ensemble development.

7.1.1 Batch Representation

Various methods to find a single representation per batch have been introduced in Chapters 5 and 6. In chapter 6 a solo context-sensitive representation per batch is presented. Experiments provided

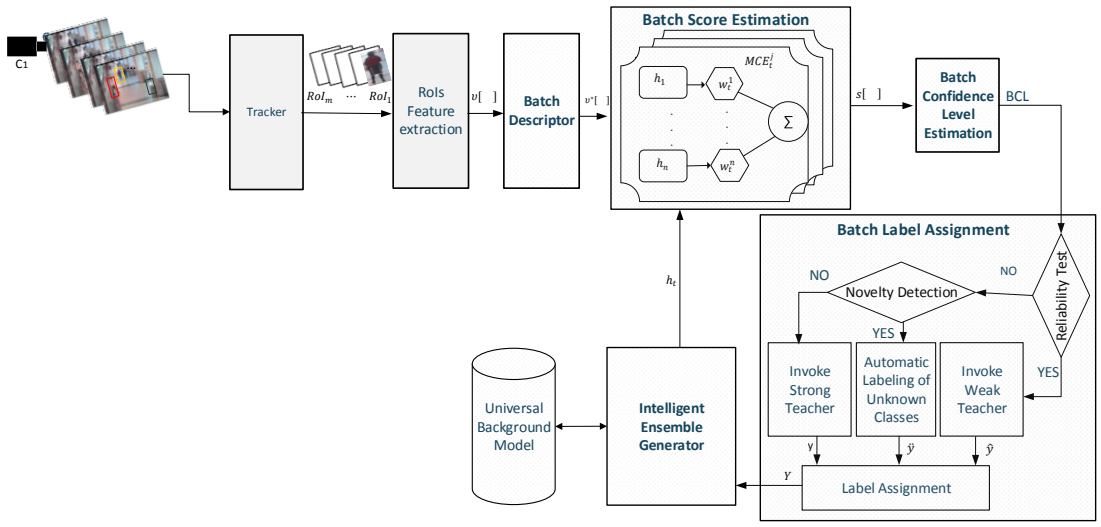


Figure 7.1: Block diagram of INEVIL

the scientific evidence for the advantage of this approach over the other group of batch-based descriptors (Chapter 5) or frame-based methods (Chapter 3). Hence context-sensitive representations have been applied in our experiments.

7.1.2 Batch Score Estimation

The learning framework is initialized to yield the same probability to every class (uniform prior), at the first place. When the features of batches of RoIs ($\upsilon_{t,f}^{*m}$) at time slot t become available, the framework starts computing the scores $\mathscr{S}(\upsilon_t^{*m}|C_k,H_{t-1})$ for every batch. The scores are obtained from the likelihood ratio test of the batch data obtained by the individual class model C_k and the Universal Background Models (as detailed in Section 4.2.1).

The composite model H_t is an ensemble of Micro-classifiers ensembles ($MCE_t^j, j = 1,...,k$). Each MCE_t^j includes classifiers that are incrementally trained (with no access to previous data) on incoming batches of j_{th} class at t, h_t^j. The individual models h_t^j are combined using a weighted majority voting, where the weights are dynamically updated with respect to the classifiers' time of design.

The prediction outputted by the composite model MCE_t^j for a given batch of ROIs (υ_t^{*m}) is

$$\mathscr{S}(C_j|\upsilon_t^{*m},MCE_t^j) = \sum_{\ell=1}^{t} W_\ell^t \mathscr{S}_\ell^j(C_j|\upsilon_t^{*m}) \tag{7.1}$$

where $\mathscr{S}_\ell^j(.)$ is the score outputted by $h_\ell^j(.)$ (the model obtained, either trained or updated, from batches of j^{th} class at timeslot ℓ), and W_ℓ^t denotes the weight assigned to model h_ℓ^j, adjusted for time t. The weights are chosen from a geometric series ($(1,...,(\beta)^\ell)$) and are updated and normalised at each time slot to give more credit to the more recent knowledge.

7.1.3 Batch Confidence Level Estimation

Various criteria have been introduced in Chapter 3 as uncertainty measures in our framework. Based on the experiments in Chapters 3 and 4, the most confident measure, defining as Eq. 7.2 outperforms other measures in our scenario.

$$\max_{C_k} \mathscr{S}(C_k|\upsilon^{*m}_t, H_{t-1}) \tag{7.2}$$

7.1.4 Batch Label Assignment

On-line learning from time-evolving environments, where labelled data is scarce, may suffer if labelling errors accumulate, which is inevitable. To mitigate this issue, the system is designed to exploit *active learning strategies*. Based on the type of oracle (teacher) available, these strategies request two groups of teachers to label the data. Strong teacher usually but not always is human and assumed to give unambiguous but expensive labels, while weak teachers provide tentative but less expensive labels. Most, but not all, weak teachers are assumed to be classification algorithms. This framework provides the opportunity to take advantage of both groups. We request the ensemble, H, to participate in learning as a weak teacher. Additionally, in order to reduce the number of the queries, we consider automating the label assignment for novel classes.

In our scenario, the number of classes is unknown beforehand. When a previously unobserved person enters the area of coverage by the camera network, the system should create a new model to represent the novel class. Applying a threshold to detect novel classes is extensively explored in the literature [91]. In INEVIL framework, if the scores associated to all observed classes $(\mathscr{S}(C_j|\upsilon^{*m}_t, H_{t-1}), j = 1,...,k)$ are significantly low (below a predetermined threshold), it is very likely that this class has not been observed before and it is considered novel and a new label ($\ddot{y}$) is automatically assigned to this/these batch(es):

$$\max_{C_k} \mathscr{S}(C_k|\upsilon^{*m}_t, H_{t-1}) < T \Rightarrow \text{data belongs to a novel class } \ddot{y}$$

Having decided that the batch data belongs to an existing class, one needs to decide which teacher to invoke. If the decision made by H is not reliable enough, that means if $\max_{C_k} \mathscr{S}(C_k|\upsilon^{*m}_t, H_{t-1}) < T_1$, a strong teacher (in this case an operator) labelling needs to be requested (y), otherwise we invoke the weak teacher and $\hat{y} = \arg\max_{C_k} \mathscr{S}(\upsilon^{*m}_t|C_k, H_{t-1})$ is assigned to the batch.

7.1.5 Intelligent Ensemble Development

An important part of our work is to propose incremental learning algorithms for never-ending scenarios in where the system can learn 24/7 from wide-area surveillance networks. As data distribution evolves in time, the framework must be able to adapt itself to track changes in order to maximize the performance. The advantage of ensembles algorithms in tackling these problems is the ability to accumulate and aggregate knowledge in the form of learned models [150].

In a typical ensemble a new classifier is build once a new set of labelled data is available. Adding new models at every time slot creates a significant limitation to the system as the computational and storage resources are limited. The target of the proposed intelligent approach is to increase the efficiency by avoiding a expansion of the ensemble size without sacrificing the performance. We explore the problem of intelligent ensemble from two different perspectives, data-level and model-level. Both algorithms exploit a "detect-and-react" strategy to detect a change either in the data distribution $p(\upsilon^{*m}_t)$ or the classifier h^j_t and then perform the subsequent adaptive procedure.

Note that, both methods exploit *universal background modelling* to create a new classifier.

7.1.5.1 Data-level Adaptation Mechanism

The first step in data-level adaptation mechanism is to inspect incoming batches of data to react to the detected change. In order to detect change in data distribution, several family of measures have been proposed in the literature that can be categorized in two main groups, signal-based and feature-based measures. The former adopts signal quality measures to accept or reject samples as well as select the best observations, modalities or classifiers. The decisions are made based on information prior to the feature extraction [34, 122]. Feature-based strategies [108] inspect the information after the feature-extraction process. Algorithm 5 details the proposed approach. In

Algorithm 5 Data-level Update

Input: $\mathscr{H}^m_{t-1}, \upsilon^{*m}_t, \upsilon^{*m}_{ref}, \forall m = 1,...,K$
Similarity assessment 7.1.5.1
$d^m_t = Similarity(\upsilon^{*m}_t, \upsilon^{*m}_{ref})$
if $d^t_m > T''$ **then**
  Adding criterion
  $h^m_t \leftarrow \upsilon^{*m}_t$
  $\mathscr{H}^m_t = (h^m_t, \mathscr{H}^m_{t-1})$
  $\upsilon^{*m}_{ref} \leftarrow \upsilon^{*m}_t$
else
  Updating a concept 7.1.5.1
  $hl^m_t = update(\upsilon^{*m}_t, h^m_{t-1})$
  $\mathscr{H}^m_t = (hl^m_t, \mathscr{H}^m_{t-1})$
end if

this paper, we focus on a novel mechanism to detect deviations in feature distributions from the reference batch of RoIs (see Figure 7.2). The framework employs a thresholding strategy to detect different levels of change and to provide an up-to-date knowledge. Once a new batch of RoIs is received, the framework assigns a label. In order to evaluate the drift level, each batch υ^{*m}_t is compared with the reference batch of each class υ^{*m}_{ref}. When a gradual change is detected, i.e. the deviation of new samples from reference batch of a given class is below a predefined threshold (T''), INEVIL will update the latest model in MCE_m with the most recent data (hl^m_t).

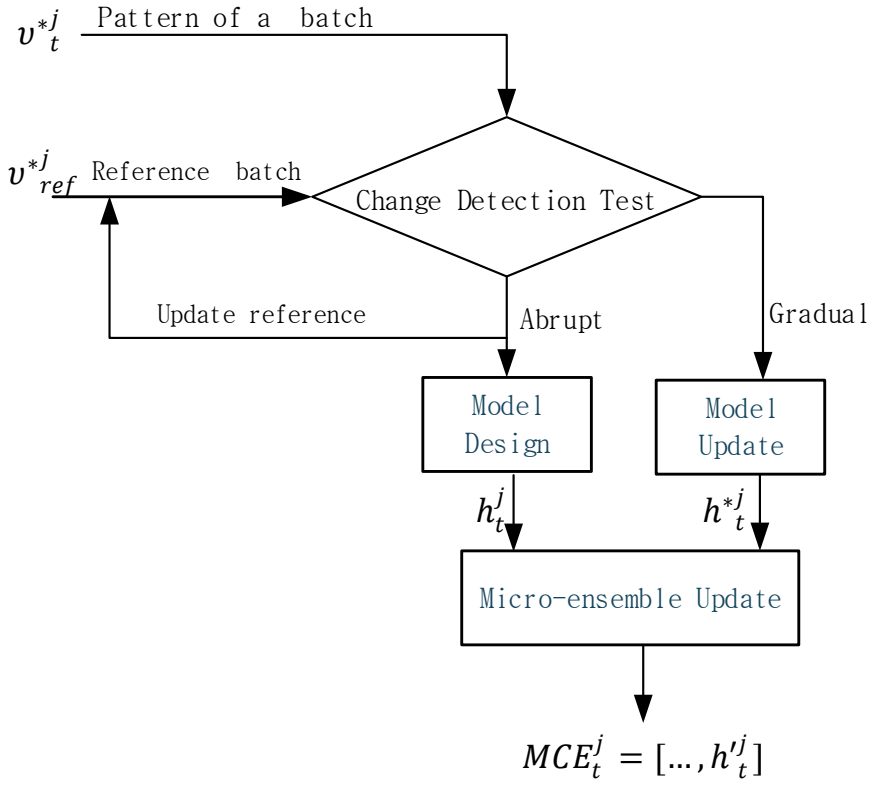


Figure 7.2: Data-level Updating Procedure

While an abrupt drift is observed (i.e. due to a new viewpoint or change of camera), the framework trains a new model using the data just captured and substitutes the recent batch as the new reference batch of RoIs.

Batch Drift Assessment

Video change detection approaches in the literature has mostly focused on the visual similarity of successive frames of the video [8, 130], while in this work batch drift assessment quantifies the change in the content of a batch of RoIs. To achieve this goal, each batch is represented by a feature vector (Section 7.2.2). Then, a pair-wise comparison between batches is employed to measure the similarity between the reference batch and the current one.

The Fisher Vector (FV) [118] was found to be one of the most effective encoding techniques in recent years [23, 140]. The similarity between two samples $\mathscr{X}$ and $\mathscr{Y}$ using the Fisher Kernel (FK) can be explicitly formulated as dot product:

$$K_{FK}(\mathscr{X}, \mathscr{Y}) = \mathscr{G}_{\geq}^{\mathscr{X}\prime} \mathscr{G}_{\geq}^{\mathscr{Y}}$$

where $\mathscr{G}_{\geq}$ is the *Fisher Vector*. More information is available at [118].

The l_2 norm is a natural norm associated with dot product, this measure is also chosen to evaluate the drift level of batches. The drift between the current batch , υ^{*m}_t, and the reference batch, υ^{*m}_{ref}, of m_{th} class, d^m_t, is formulated as:

$$d^m_t = \langle \upsilon^{*m}_t, \upsilon^{*m}_{ref} \rangle \tag{7.3}$$

To the best of our knowledge this is the first attempt to apply this measure for batch change detection. Favourable results (see 7.2) indicate the effectiveness of this light yet effective measure.

Updating GMMs with new observations

Once gradual drift is observed, the data from the batches predicted to belong from the same class is used to generate the class model by *tuning of the (h^m_{t-1}) parameters*, in a maximum *a posteriori* (MAP) sense. The rationale behind this method is basically similar to updating the individual models for UBM. The adaptation process consists of two main estimation steps. First, for each component of the h^m_t, a set of sufficient statistics is computed from a set of B class specific feature vectors, $\upsilon^{*m}_t = \{\mathbf{x}_1, ..., \mathbf{x}_B\}$ computed from the batch data:

$$n_i = \sum_{b=1}^{B} p(i|\mathbf{x}_b) \tag{7.4}$$

$$E_i(\mathbf{x}) = \frac{1}{n_i} \sum_{b=1}^{B} p(i|\mathbf{x}_b)\mathbf{x}_b \tag{7.5}$$

$$E_i(\mathbf{x}\mathbf{x}^t) = \frac{1}{n_i} \sum_{b=1}^{B} p(i|\mathbf{x}_b)\mathbf{x}_b\mathbf{x}^t_b \tag{7.6}$$

where $p(i|\mathbf{x}_b)$ represents the probabilistic alignment of $\mathbf{x}_b$ into each h^m_{t-1} component. Each h^m_{t-1} component is then adapted using the newly computed sufficient statistics, and considering diagonal covariance matrices. The update process can be formally expressed as:

$$\hat{w}_i = [\alpha_i n_i/B + (1-\alpha_i)w_i]\,\xi \tag{7.7}$$

$$\hat{\boldsymbol{\mu}}_{\boldsymbol{i}} = \alpha_i E_i(\mathbf{x}) + (1-\alpha_i)\boldsymbol{\mu}_{\boldsymbol{i}} \tag{7.8}$$

$$\hat{\Sigma}_i = \alpha_i E_i(\mathbf{x}\mathbf{x}^t) + (1-\alpha_i)(\boldsymbol{\sigma}_{\boldsymbol{i}}\boldsymbol{\sigma}^t_{\boldsymbol{i}} + \boldsymbol{\mu}_{\boldsymbol{i}}\boldsymbol{\mu}^t_{\boldsymbol{i}}) - \hat{\boldsymbol{\mu}}_{\boldsymbol{i}}\hat{\boldsymbol{\mu}}_{\boldsymbol{i}}{}^t \tag{7.9}$$

$$\boldsymbol{\sigma}_{\boldsymbol{i}} = \mathrm{diag}(\Sigma_i) \tag{7.10}$$

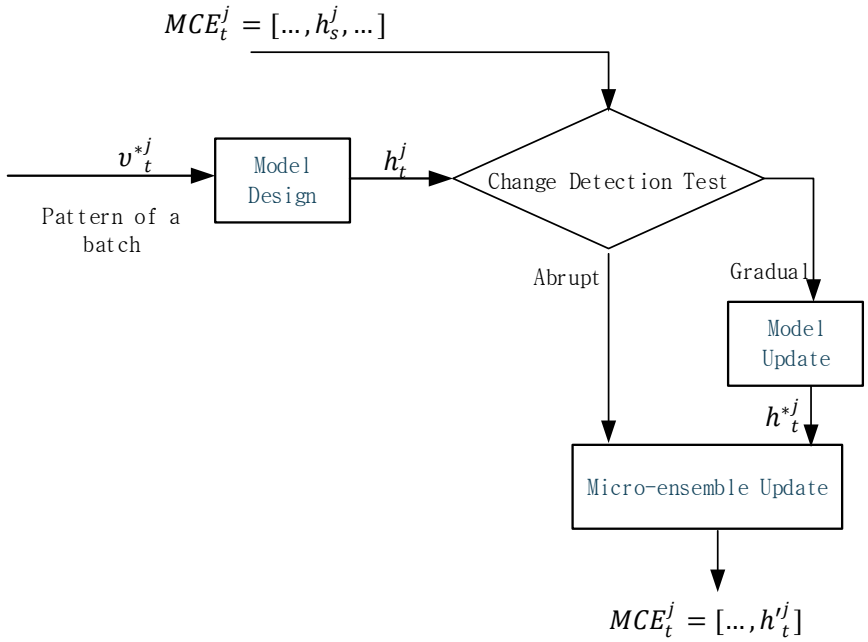


Figure 7.3: Model-Level Adaptation Mechanism Procedure

where $\{w_i, \boldsymbol{\mu_i}, \boldsymbol{\sigma_i}\}$ are the original h_{t-1}^m parameters and $\{\hat{w}_i, \hat{\boldsymbol{\mu}}_{\boldsymbol{i}}, \hat{\boldsymbol{\sigma}}_{\boldsymbol{i}}\}$ represent their adaptation to the specific class. To assure that $\sum_i w_i = 1$ a weighting parameter ξ is introduced. The α parameter is a data-dependent adaptation coefficient. Formally it can be defined as:

$$\alpha_i = \frac{n_i}{r + n_i} \tag{7.11}$$

The relevance factor r weights the relative importance of the original values and the new sufficient statistics.

7.1.5.2 Model-level Adaptation Mechanism

In this section, we propose a mechanism for the development of the ensemble with higher potential to identify recurrent concept drift. The framework employs change detection strategies at the learner level in order to (re-)identify concept drifts aiming at provoking a favourable yet less involved response.

"It is desired that the individual learners should be accurate and diverse" [155]. Although there is no well-accepted definition for diversity in the classifier ensemble field, there is a general agreement on the qualitative notion of diversity and its role on the performance of an ensemble [3]. An ensemble consisting in a set of identical classifiers does not bring any advantage while maintaining a good diversity with minimum number of classifiers is desirable.

We propose an algorithm that keeps a limited number of the classifiers inside the ensemble without sacrificing the performance. Algorithm 6 outlines our approach. The framework receives multiple micro-ensembles (H_t^m). Once the batch(es), υ_t^{*m}, in time slot t classified as class "m"

Algorithm 6 Model-Level Adaptation Mechanism

Input: $\mathscr{H}_{t-1}^m, h_t^m$
Similarity assessment
$d_{tj}^m = Similarity(h_j^m, h_t^m), \forall j = 1, ..., K$7.1.5.2
Closest models
$\exists k \in 1, ..., K, d_k^t < d_j^t, \forall j \neq k$
if $d_k^t > T'$ **then**
Adding criterion
$\mathscr{H}_t^m = (h_t^m, \mathscr{H}_{t-1}^m)$
else
Updating a concept
$h\prime_t^m = merge(h'^m_k, h_t^m)$ 7.1.5.2
$h_k^m = []$
$\mathscr{H}_t^m = (h\prime_t^m, \mathscr{H}'^m_{t-1})$
end if

(either by invoking teachers or automatically), the framework trains a model (h_t^j). The similarity scores d_{tj}^m are computed between all the models available in the corresponding micro-ensemble ($MCE_j^{(t-1)}$) and the newly trained model (h_j^t). Once similarity scores are obtained, we searched over all the models to find the one with minimum distance with the recent one.

If $d_{tk}^m = \min d_{tj}^m, j = 1, ..., K$ (K is the number of models inside m_{th} micro-ensemble) is high enough (above a predefined threshold, (T'), the model is distinct enough, thus it will be added to the ensemble, otherwise the closest pair is merged into a single model ($\hat{h}_t^m$).

Hence, we need a notion of similarity between models as well as a strategy for updating the learners.

Model-level Drift Assessment

The Kullback-Leibler divergence is a natural similarity measure between two distributions. Although it cannot be analytically computed for GMMs, an efficient and accurate approximation of KL-divergence for GMMs is proposed in [53]. Assume $h(x) = \sum_{i=1}^n \alpha_i h_i(x)$ and $h'(x) = \sum_{k=1}^m \beta_k h'_k(x)$ are two Gaussian Mixture densities whose KL-divergence we want to compute. Generally, the KL-divergence between two GMMs can be approximated by:

$$KL(h \parallel h') \approx \sum_{i=1}^{n} \alpha_i min_{k=1}^m KL(h_i \parallel h'_k) \tag{7.12}$$

The approximation is based on a matching function between each element of h and an element of h' that is the most similar to it. Various methods including the Hungarian algorithm have been employed to find corresponding components.

Since, in GMM-UBM, the GMMs are obtained from a maximum a posteriori adaptation of a universal background model, the both densities have the same number of components and there is a well justified correspondence between components, the KL-divergence can be approximated as:

$$KL(h \parallel h') = \sum_i \alpha_i KL(h_i \parallel h'_i) \tag{7.13}$$

where the KL-divergence between components $h_i(\mu_1, \Sigma_1)$ and $h'_i(\mu_2, \Sigma_2)$ can be formulated as:

$$KL(h_i \parallel h'_i) = \frac{1}{2}(log\frac{|\Sigma_2|}{|\Sigma_1|} + Tr(\Sigma_2^{-1}\Sigma_1) + \big(\mu_1 - \mu_2\big)^T \Sigma_2^{-1}(\mu_1 - \mu_2) - d\big) \tag{7.14}$$

Finally, the distance between two distributions h and h' is computed as:

$$d_{hh'} = \frac{KL(h \parallel h') + KL(h' \parallel h)}{2} \tag{7.15}$$

Updating a Concept

The problem of updating GMM has mostly appeared in situations where: 1) a Gaussian mixture is fitted but the mixture components are not separated enough [61, 117, 138], 2) in presence of non stationary environment the model is not representative as time passes and the Gaussian mixture model needs to be updated to track environment change [72, 85]. These algorithms mainly focus on updating a GMM by either merging or splitting the components, while combining two GMMs is a less explored area.

We present a method for incrementally merging GMMs that avoids the necessity to keep all data points. Suppose we have learned two GMMs from two sets of observations $f(x)$ and $g(x)$ with n' and n'' points:

$$f(x) = \sum_{i=1}^{M} f_i(x'_i, \pi'_i, \mu'_i, \Sigma'_i)$$

$$g(x'') = \sum_{j=1}^{M} g_j(x_j{}'', \pi_j{}'', \mu_j{}'', \Sigma_j{}'')$$

, where each component is represented by its weight (π), mean (μ), and covariance (Σ).

We start by finding corresponding components in different GMMs (finding the closest component from two or multiple GMM). We present the method for a pair of 2-component GMMs, while the method can be extended for multiple GMM. Suppose, the i_{th} component of $f(x)$ corresponds to j_{th} component of $g(x)$, that can be merged and form the k_{th} component of $m(x) = \sum_{j=1}^{M} m_k(x_k, \pi_k, \mu_k, \Sigma_k)$. Note that the number of points in j_{th} component is expected to be equal to the product of the component weight ($\ddot{\pi}_j$) and the total number of points in the GMMs ($\ddot{n}$). Using the definition of mean, variance and prior we derive:

$$\pi_k = \frac{n'\pi'_j + n''\pi''_j}{n' + n''} \tag{7.16}$$

$$\mu_k = \frac{\sum x_k}{n} = \frac{\sum x'_i + \sum x''_j}{n'\mu'_i + n''\mu''_j} = \frac{n'\pi'\mu'_i + n''\pi''_j\mu''_j}{n'\pi'_i + n''\pi''_j} \tag{7.17}$$

$$\Sigma_k = E(x_k{}^2) - E^2(x_k) = \frac{n'\pi'(\Sigma_i + \mu'_i\mu'^T_i) + n''\pi''_j(\Sigma''_j + \mu''_j\mu''^T_j)}{n'\pi'_i + n''\pi''_j} - \mu_k\mu_k{}^T \tag{7.18}$$

7.2 Experimental Methodology

7.2.1 Experimental Setup

Experiments were conducted on public indoor (CAVIAR [112], SAVIT-SOFTBIO [15]) and outdoor (PETS) datasets. Seven scenarios of CAVIAR (*OneLeave ShopReenter1, Enter ExitCrossingPaths1, OneShopOneWait1, OneStop Enter2, WalkBy Shop1front*) as well as two views of scenario S2.L1 of PETS2009 have been applied in our experiments. We also carried out some experiments on three subsets of SAVIT-SOFTBIO: 1) *SAVIT*. 2) *SAVIT Non-Over*. 3) *SAVIT Recurrent*.

To extract the RoIs, we employed an automatic tracking approach [135] to track objects in the scene and generate streams of bounding boxes, which define the tracked objects' positions. As the tracking method fails to perfectly track the targets, a stream may include RoIs of distinct objects.

7.2.2 Video Representation

RoI Representation

Our reference image descriptor is an improved version of FV, since the FV was found to serve as the most effective encoding technique for pooling approaches in recent studies [25]. Given an image (RoI), the IFV υ is obtained by extracting a dense collection of patches and corresponding local image features (herein, SIFT [89]) from the image at multiple scales. Each descriptor ($x_i \in R^D$) is then soft-quantized using a Gaussian Mixture Model with K components. To avoid the curse of dimensionality, Principle Component Analysis (PCA) is applied to the full set of features as a pre-processing step. The number of features in each stream is reduced to 200 dimensions.

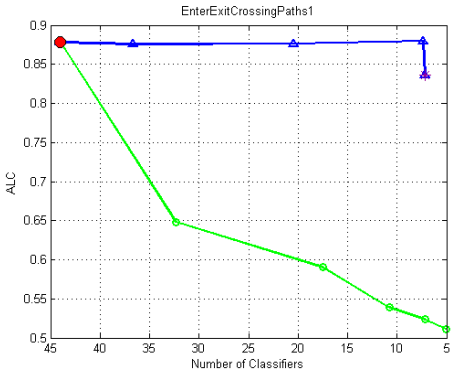
Batch Representation

Once the RoI wise features are extracted, a weighted average over time, which gives more credit to more trustworthy RoIs, is adopted to produce a batch representation (consider Chapter 6 for more details).

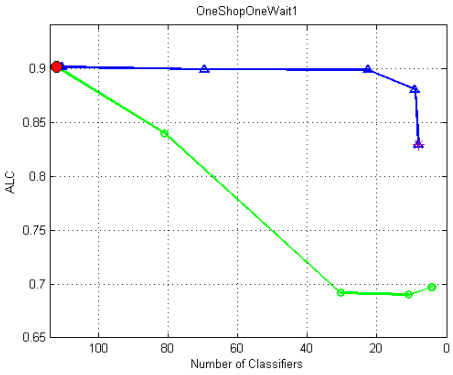
7.2.3 Baseline Methods

The work closest in spirit to INEVIL is [58], that proposed a never-ending framework for one dimensional real value time series. Since, we deal with multiple high-dimensional data streams, the framework is not applicable in our scenario. We compare INEVIL framework with three baseline approaches: 1) Ensemble Classifier Model, that add and modify members of an ensemble. This

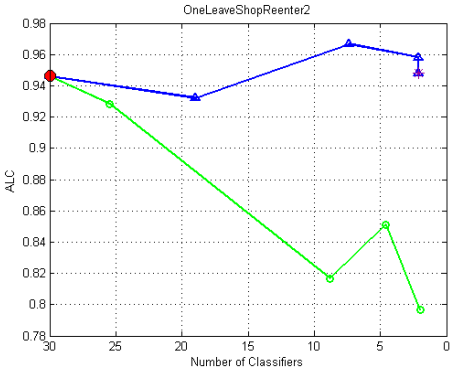
Figure 7.4: Some streams of SAIVOT (Timeslots when a new model is added to the system are highlighted in the sequences)



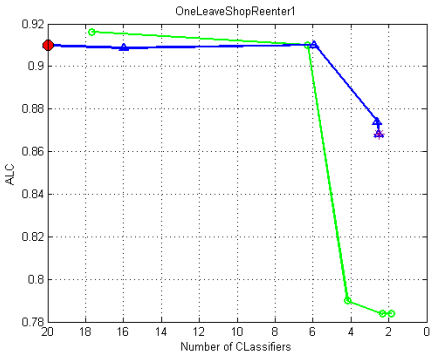
(a) EnterExitCrossingPaths1



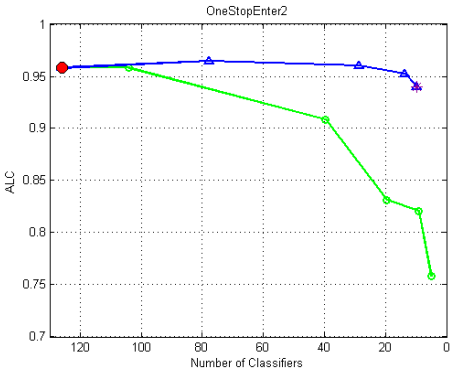
(b) OneShopOneWait1



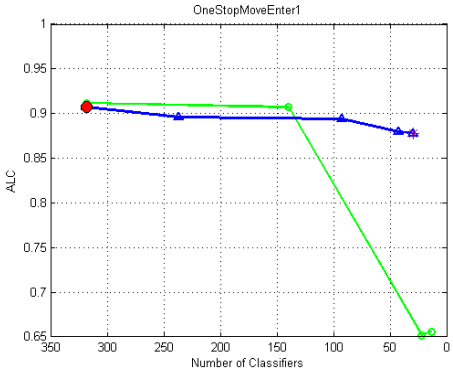
(c) OneLeaveShopReenter2



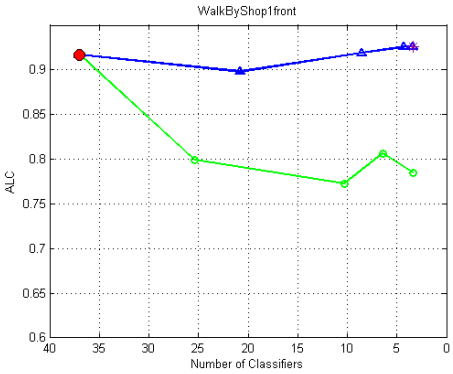
(d) OneLeaveShopReenter1



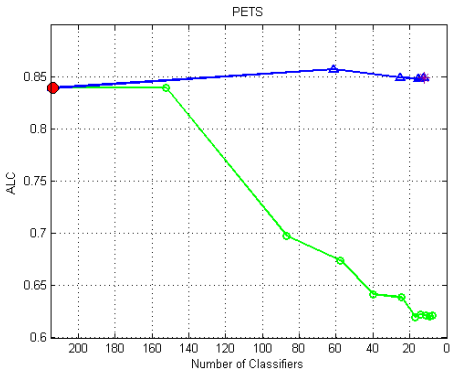
(e) OneStopEnter2



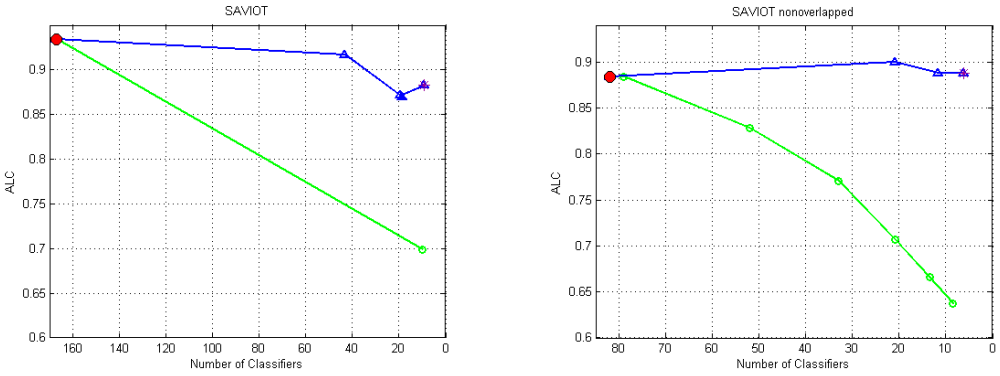
(f) OneStopMoveEnter1



(g) WalkByShop1front



(h) PETS



(i) SAVIOT (j) Non-overlapped

Figure 7.5: Comparison of the performance of feature-level and model-level intelligent ensemble development on real-world datasets (ALC against number of classifiers in descending order). The signs denote the results of AA, INEVIL, NEVIL.ubm, and Incremental Learning, respectively.

method adds a new member to the ensemble as new data arrives. 2) Incremental methods (single classifier models): at the other side of extreme these methods perform a continuous adaptation of the model once new observations received. 3) Active Add: between two extremes *active adaptation* adds a new model whenever an abrupt change is detected. These methods are detailed as follows.

Ensemble Classifier Models: NEVIL.ubm

NEVIL.ubm [76] is an example of the recent learning algorithms for learning from multiple unregulated streams in a non-stationary environment where both concept drift and concept evolution are available. The framework is detailed in Chapter 4. NEVIL.ubm trains a new classifier when new batches are available. Despite a time-weighted strategy to control the complexity of the framework, the size of the ensemble grows over time.

Single Classifier Model: Incremental Learning

Incremental learning performs a continuous adaptation of the model once a new observation is received. The composite model H_t includes only one classifier per class that is incrementally trained (with no access to previous data) on incoming batches of j_{th} class at t, h_t^j. The adaptation process for data-level and model-level is performed using the techniques proposed in sections 7.1.5.1, 7.1.5.2, respectively.

Active Methods: Active Add

The *Active Add* is based on change detection mechanisms (proposed in sections 7.1.5.2, and 7.1.5.1). Once an abrupt change is detected, a new classifier is built and added to the corresponding micro-ensemble (MCE_t^j). The framework only reacts to abrupt changes and no update is performed in case of gradual drift.

7.3 How well do intelligent updating mechanisms work under a never-ending setting?

We evaluate the INEVIL under the never-ending setting. Active mechanisms are employed to address long-term monitoring at two different levels, data-level and model-level.

7.3.1 Effectiveness of the Data-level Adaptation

7.3.1.1 Video Drift Assessment Results

We proposed a criterion for visual data change assessment. The timeslots when flag is triggered and a new model is added to the ensemble are highlighted in Figure 7.4. We observe that abrupt change in illumination and pose may trigger the flag. We applied this criterion to track the changes and perform an appropriate adaptation.

7.3.1.2 Data-level Adaptation

To evaluate the effectiveness of the adaptation algorithm on size and accuracy of learning system, we compared our method with three baseline approaches. Figure 7.5 illustrates the comparative results across baseline approaches on multiple video datasets as a function of number of classifiers from which we can observe that: a) keeping all the classifiers (red points) does not bring an advantage for the system. b) Only track and re-act to abrupt change (AA method) do not bring any advantage to the framework. c) While using wise update and add strategy, INEVIL performs as well as and in most cases better that conventional ensembles by keeping only a limited number of classifiers. The number of the learners is a function of number of classes that have been observed at the scene. For example, the system obtained 90% ALC (which is the best ALC obtained for this set) by keeping 21 models for the 7 classes present at "SAIVT-NonOver" dataset. The average cost is 3 models per person. However the cost increased for more occluded datasets (e.g. PETS with average cost of 6 classifiers per classes), still the framework controls a dramatic expansion of the size of the models without sacrificing or in cases (e.g. PETS, SAIVT-NonOver, WalkbyShop1Front, OneShopOneWait1, EnterExitCrossingPath1, OneLeaveShopReenter1) even improving the performance.

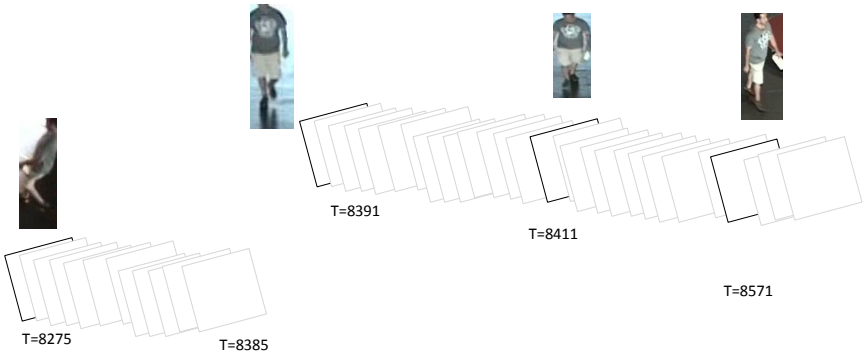
7.3.2 Effectiveness of the Model-level Adaptation

We evaluate INEVIL on the task of long-term tracking using model-level assessment strategy. The performance of the system is evaluated using ALC versus the number of classifiers.

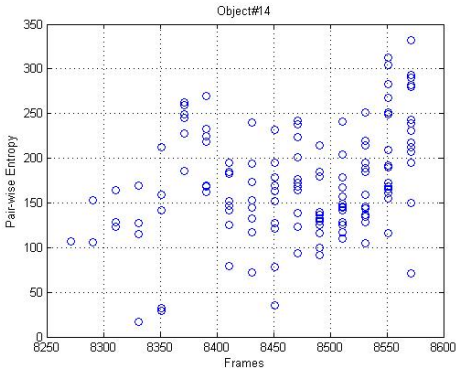
7.3.2.1 Model Change Detection

In Figure 7.6, the timeslots when a flag triggered and a new model is added to the ensemble are highlighted. We observe that abrupt change in illumination, pose may trigger the flag. Pair-wise distance between models inside an ensemble is applied as notion of diversity in this framework.

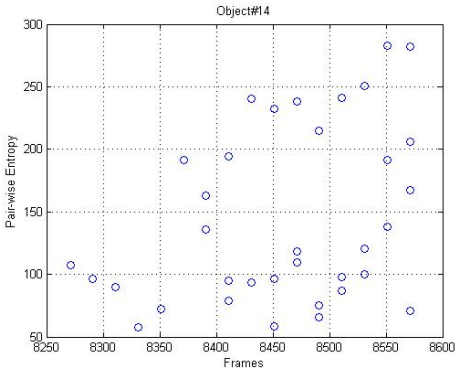
Figure 7.6: Changes in illumination and pose triggered adding models (These points are highlighted in the sequences)



(a) streams correspond to object #14



(b) Pair-wise entropy of the models present inside MCE #14 using NEVIL.ubm



(c) Pair-wise entropy of the models #14 using INEVIL

Figure 7.7: An example of micro-ensemble diversity using Model-Level adaptation mechanism

Figure 7.7 shows an example of the the micro-ensemble diversity corresponds to subject #14 using NEVIL.ubm (Figure 7.7b) and INEVIL (Figure 7.7c) over time(horizontal axis denotes the frame number). Although the number of the models has been dramatically reduced using INEVIL (Figure 7.7c), diversity range (difference between the minimum and maximum distance) has not been changed. Figure 7.8 illustrates the performance of model-drift assessment strategy on various video clips. The ALC is presented as a function of the number of classifiers. We observe that in 7 datasets (i.e. SAVIT, SAVIT Nonover, Enter ExitCrossingPaths1, OneShopOneWait1, OneStop Enter2, WalkBy Shop1front) out of 9, the framework obtains comparable performance by wise development of only 20% number of the maximum classifiers possible. The average number of classifiers per person is 1.8 (21 models for 11 classes). With only one exception (OneExitCrossingPath1) INEVIL outperforms Incremental learning. AA provides the poorest performance confirming the importance of on updating procedure.

7.3.2.2 Which Adaptation Mechanism Benefits our Setting Better?

Performance

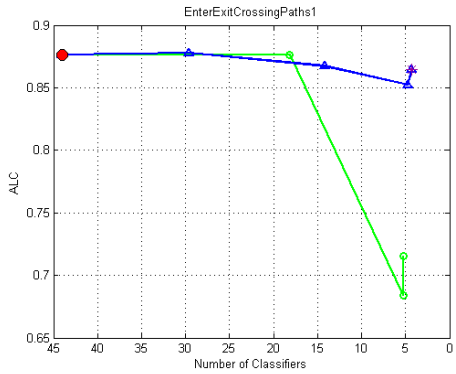
Figure 7.9 presents the results of both levels of adaptation on various video datasets as a function of the number of classifiers. Since, there is not a single size that everyone agrees as the most reasonable cost, we compare the performance of methods with 20% of classifies. Model-level adaptation is specially designed for addressing recurrent drift outperforms inwhere recurrent drift is present (i.e SAVIT, SAVIT-recurrent, SAVIT Non-Over). However, in 6 out of 11 datasets the data-level adaptation provides better results.

Time Efficiency

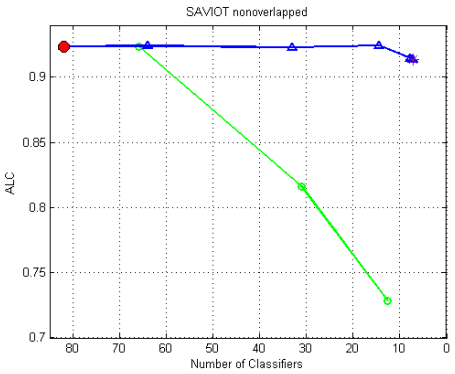
Since INEVIL was developed in MATLAB without any efficiency concerns (running in an Intel Core i7 at 3.2GHz), a straightforward assessment of the time efficiency is not adequate. Figure 7.10a depicts INEVIL processing time for SAVIT-ReCurrent as a function of frame number. For a framerate of 25fps, one second is spanned by the batch in our experiments. The analysis time grows naturally with the complexity of the dataset; however feature-level adaptation delivers faster decisions. However, for both methods the maximum processing time of a second video is 0.25 second. Thus, the INEVIL framework is able to process in real time.

Memory

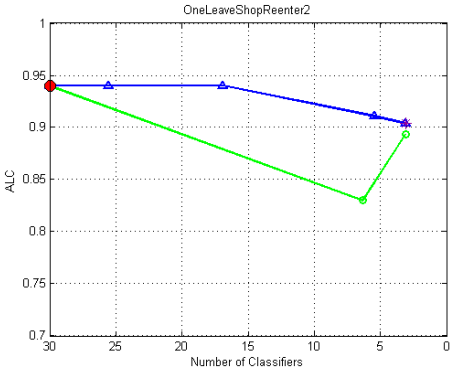
Figure 7.10b illustrates the number of classifiers inside ensemble over time. Once a new concept or an abrupt drift is detected, a new model is added to the ensemble. The dataset is organized in the way that the last dramatic drift is observed at frame=1700, when is the last time a new concept is observed. We observe that the model-adaptation method experiences a stability of size after this point while the size of INEVIL in feature based algorithm increases slightly.



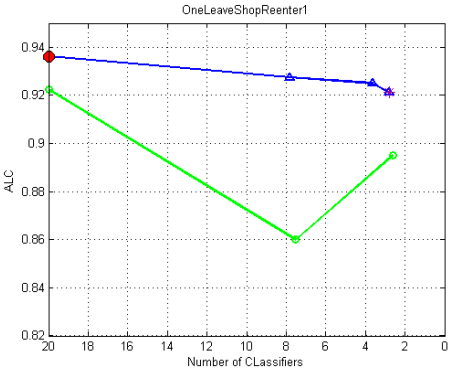
(a) EnterExitCrossingPaths1



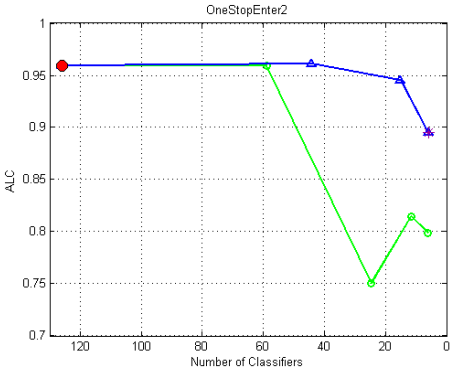
(b) Non-overlapped



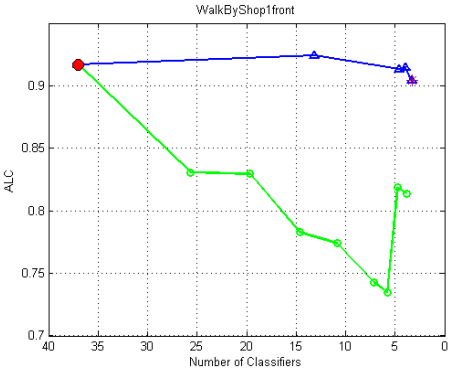
(c) OneLeaveShopReenter2



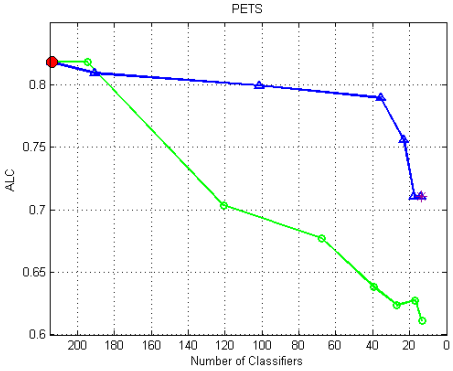
(d) OneLeaveShopReenter1



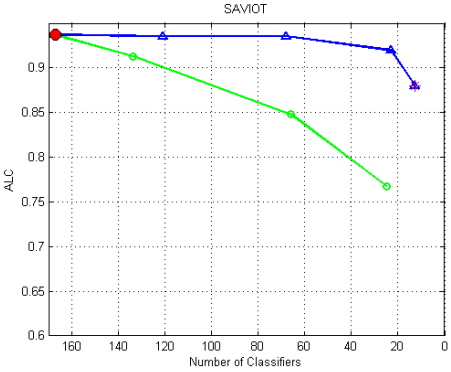
(e) OneStopEnter2



(f) WalkByShop1front

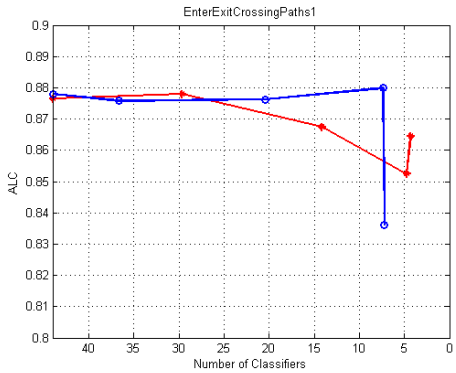


(g) PETS

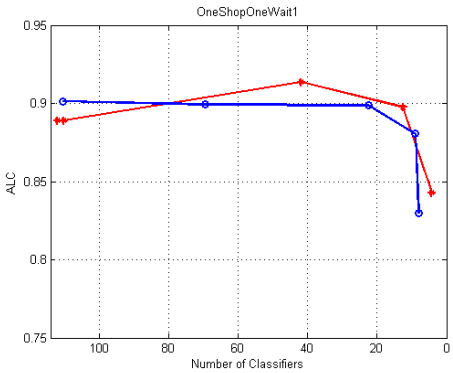


(h) SAVIOT

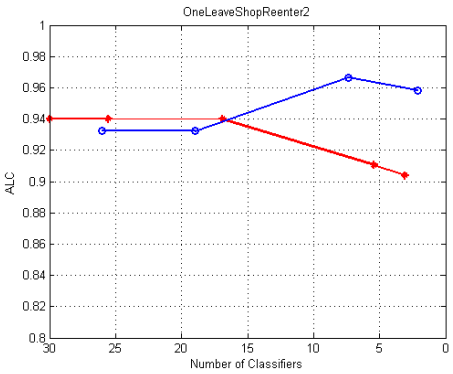
Figure 7.8: Comparison of the performance of INEVIL against multiple baseline approaches on real-world datasets (ALC against number of classifiers in descending order). The signs denote the results of AA, INEVIL, NEVIL.ubm, and Incremental Learning, respectively.



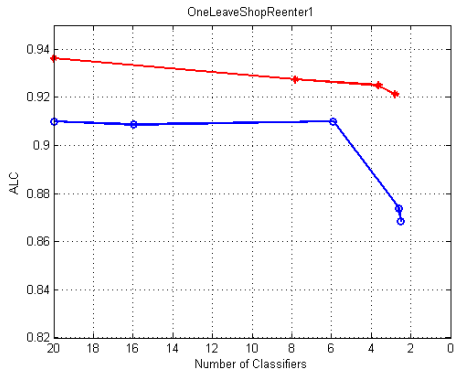
(a) EnterExitCrossingPaths1



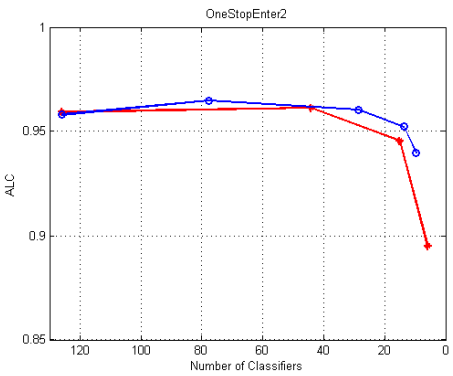
(b) OneShopOneWait1



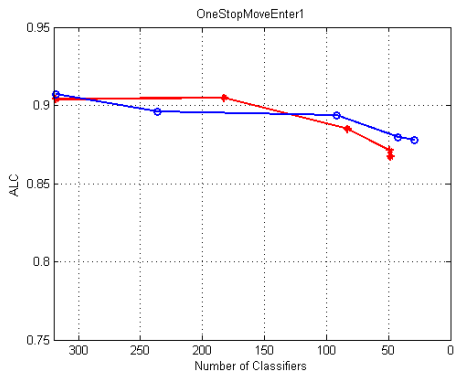
(c) OneLeaveShopReenter2



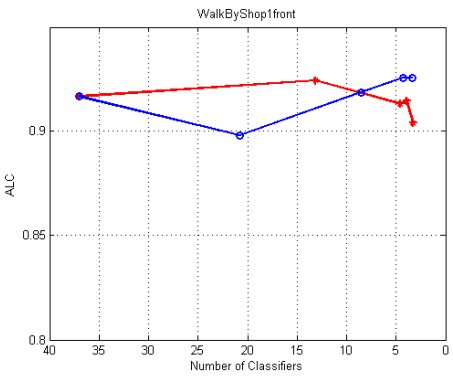
(d) OneLeaveShopReenter1



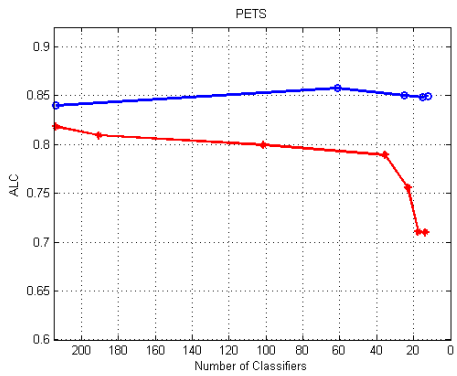
(e) OneStopEnter2



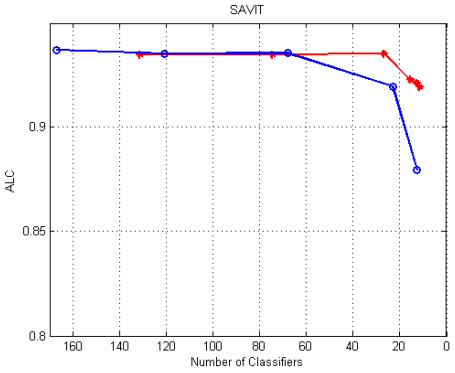
(f) OneStopMoveEnter1



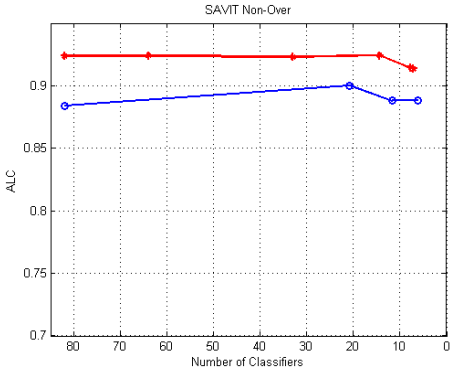
(g) WalkByShop1front



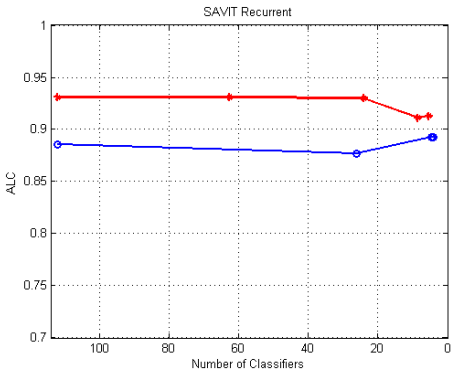
(h) PETS



(i) SAVIOT

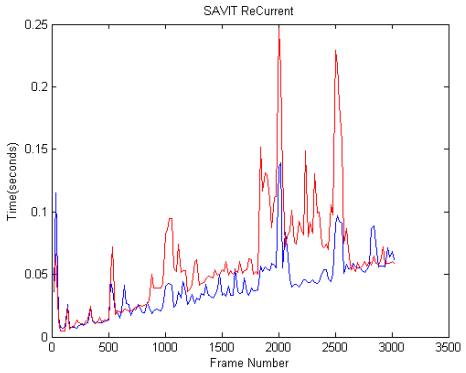


(j) SAVIT Non-Over

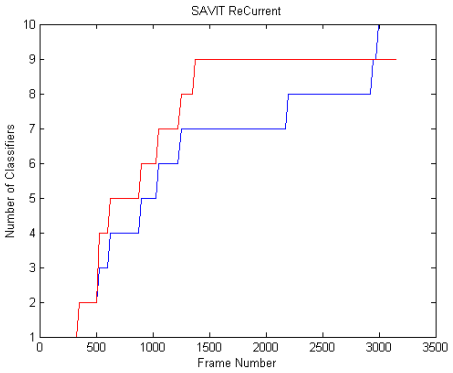


(k) SAVIT Recurrent

Figure 7.9: Comparison of the performance of data-level and model-level intelligent ensemble development on real-world datasets (ALC as a function of the number of the classifiers in descending order). The signs denote the results of feature-level and model-level, respectively.



(a) Time efficiency



(b) Memory

Figure 7.10: Comparison of the performance of data-level and model-level intelligent ensemble development on SAVIT ReCurrent in terms of processing time and memory over time. The signs denote the results of feature-level and model-level, respectively.

7.4 Discussion

In this chapter, we present a learning setting yet unexplored in the literature but with wide relevance, especially in long-term person re-identification over multiple video cameras. We proposed an adaptive ensemble that wisely develops over time with respect to drift level in order to reflect the latest concepts as well as controlling the complexity of the system.

The framework receives directly the tracked sequences outputted by the tracking system and maintains a global object identity common to all the cameras in the system. We proposed a criterion to find more trustworthy RoIs. We further combine the information of RoIs to output a single representation vector per batch. The results indicate that it is worthwhile to give more credit to better RoIs. INEVIL adopts a UBM-normalized strategy in a class-based ensemble, where an individual ensemble (so called micro-ensemble) is trained for every single class. The framework monitors the change either in data distribution or classifier.

We proposed a simple yet effective data change detection that triggers a proper adaptation mechanism to react to the detected change either by updating an existing model or adding a new learner to the ensemble. The results indicate that we obtain a high and stable ALC (> 0.85) only by keeping average 3 models per class. Change in illumination and pose trigger building new models, while in case of gradual drift an existing classifier is updated by current observation.

The data-level adaptation effectively controls the size of the ensemble without sacrificing the performance. However when a concept drift reappears after a time, it may add new models to the ensemble. To get further closer to a practical solution, model-level adaptation was proposed which has a better capability (better performance with less complexity) to address recurrent concept drift. Favourable results on datasets with recurrent drift indicate the effectiveness of this method.

Both frameworks (either data or model level monitoring) share the same characteristics that make them effective for real-world applications: 1) adapting the knowledge using very few observations, in fact the each batch is translated into a single feature vector. 2) interactively invoke the oracles to stay on track. 3) on-line learning from streams without access to previous data.

Chapter 8

Conclusion

In this thesis, we focused on unsupervised long-term tracking over multiple distributed cameras, where people cross the FoV of multiple cameras over time and cameras monitor the environment for unbounded-time. The problem of open-world human re-identification was seen from a time-series mining perspective where visual streams are endlessly generated. First, a new learning concept was formulated and a desirable learning methodology was proposed. We proposed multiple frameworks that were developed over time, getting more efficient, effective and accurate. We conducted several experiments to assess the stability, performance and complexity of the proposed frameworks.

In the first level, we proposed NEVIL that utilizes *discriminative ensembles* to actively classify streams. While having the benefit of a robust classifier design, it was difficult to detect the novel classes. Besides, the framework got biased towards the majority class in the case of severe class imbalance. To address these problems, the NEVIL framework was extended with *generative models* within a new framework, allowing a double threshold strategy to detect novel classes and avoiding the pitfall of classifying every batch as the majority class. Hence, in the second phase of this research two novel frameworks were proposed that employ class-based ensembles. As one of the most popular generative approaches, the use of Gaussian Mixture Models to train models seemed a natural choice. The GMMs-based framework, named as NEVIL.gmm, delivered superior performance compared to NEVIL, however, stability in high-dimensional visual data is still a big issue and the novel class detection is unreliable due to the difficulty of setting a suitable threshold. NEVIL.ubm adopts a *UBM- normalized* strategy resulting in better and faster decisions as well as a robust and precise classifiers' parameter estimation, even when only a small amount of data (here, a single shot) is available. From the performance perspective, this framework has shown promising performance with a fairly little human collaboration. However, growing complexity in terms of runtime and memory is still the main concern for long-term observation.

The last framework, INEVIL, was specially designed for long-term tracking. We proposed an adaptive ensemble that wisely develops over time with respect to drift level in order to reflect the latest concepts as well as controlling the complexity of the system. We studied the system from both data-level and model-level. Both methods effectively control the size of the ensemble

without sacrificing the performance, however the former unleashed better potential to address recurrent drift.

A second line of study concerned with the representation of a batch of RoIs. Firstly, two spatio-temporal fusion strategies, which do not necessary need a controlled environment, to combine the patterns of RoIs in a given batch are presented. Then, the winner method is generalized to output a context-sensitive batch representation. An unsupervised criterion was proposed to score RoIs. Experiments indicated that it is worthwhile to consider the context information and give more credit to more trustworthy RoIs.

Benefits of this thesis take the first steps towards an open-world person re-identification over multiple distributed cameras during an unlimited time frame. Numerous experiments on various datasets provide the evidence of the effectiveness of the proposed algorithms.

Future Work

The studies of this thesis, although with some conclusive results, constitute the starting point for a possible larger project, with focus on learning algorithm from evolving environment as well as visual data representation schemes.

First line concerns with the introduction of a new or modification of a current version of the learning algorithms. These methods are listed as follows: 1) Inspired by probability models proposed in [120] for open-set recognition, we will investigate the impact of such models on the performance of the proposed framework for an open-set person re-identification problem. 2) There has been a keen interest on zero-shot learning methods for video analysis [52]. Exploring these approaches in our scenario seems an interesting field of study. 3) Never-ending learning is still a quite young (back to 2010 [20]) and less explored area in machine learning field. In this thesis, we tried to cover accuracy, knowledge and efficiency domains. However, there is still room for improvement; exploring more accurate and efficient methods constitute the future direction of our study.

Second line will focus on the study of new representation schemes. The NSF report on Future Challenges for the Science and Engineering of Learning listed depth as one of the missing ingredients [12]. Since, deep learning methods have shown their incredible potential to boost the performance and robustness of a system in various studies, we intend to exploit such approaches to learn a robust and effective model.

Finally, video analysis is a multi-disciplinary field, related to different areas. It is expected that combining different but complementing strategies, particularly never-ending learning approaches and deep-learning methods, increases the performance of the framework, an area which has not been explored yet.

Despite all the progress in the field of artificial intelligence, design a fully-automatic surveillance system is still in infancy stages. Exploring aforementioned ideas may be a step, however small, forward.

References

[1] M. Abouelenien, Y. Wan, and A. Saudagar. Feature and decision level fusion for action recognition. pages 1–7, July 2012.

[2] M. R. Ackermann, C. Lammersen, M. Märtens, C. Raupach, C. Sohler, and K. Swierkot. StreamKM++: A clustering algorithms for data streams. In *Journal of Experimental Algorithmics*, pages 173–187, 2010.

[3] M. A. O. Ahmed, L. Didaci, G. Fumera, and F. Roli. An empirical investigation on the use of diversity for creation of classifier ensembles. In *Multiple Classifier Systems - 12th International Workshop, MCS 2015, Günzburg, Germany, June 29 - July 1, 2015, Proceedings*, pages 206–219, 2015.

[4] T. M. Al-Khateeb, M. M. Masud, L. Khan, and B. Thuraisingham. Cloud guided stream classification using class-based ensemble. In *Proceedings of the 2012 IEEE Fifth International Conference on Cloud Computing*, CLOUD '12, pages 694–701, Washington, DC, USA, 2012. IEEE Computer Society.

[5] A. Alahi, R. Ortiz, and P. Vandergheynst. Freak: Fast retina keypoint. In *Proceedings of IEEE Conference on Computer Vision And Pattern Recognition*, pages 510–517, 2012.

[6] L. A. Alexandre, A. C. Campilho, and M. Kamel. On combining classifiers using sum and product rules. *Pattern Recognition Letters*, 22(12):1283–1289, 2001.

[7] C. Alippi, G. Boracchi, and M. Roveri. Just-in-time classifiers for recurrent concepts. *IEEE Trans. Neural Netw. Learning Syst.*, 24(4):620–634, 2013.

[8] E. E. Apostolidis and V. Mezaris. Fast shot segmentation combining global and local visual descriptors. In *IEEE International Conference on Acoustics, Speech and Signal Processing, ICASSP 2014, Florence, Italy, May 4-9, 2014*, pages 6583–6587, 2014.

[9] R. Bastos. *FIRST - Fast Invariant to Rotation and Scale Transform: Invariant Image Features for Augmented Reality and Computer Vision*. VDM Verlag, Saarbrücken, Germany, Germany, 2009.

[10] H. Bay, A. Ess, T. Tuytelaars, and L. Van Gool. Speeded-up robust features (surf). *Comput. Vis. Image Underst.*, 110(3):346–359, June 2008.

[11] N. Begum and E. Keogh. Rare time series motif discovery from unbounded streams. *Proc. VLDB Endow.*, 8(2):149–160, Oct. 2014.

[12] I. G. Y. Bengio and A. Courville. *Deep Learning*. 2016. Book in preparation for MIT Press.

[13] J. Berclaz, F. Fleuret, E. Türetken, and P. Fua. Multiple object tracking using k-shortest paths optimization. *IEEE Trans. Pattern Anal. Mach. Intell.*, 33(9):1806–1819, 2011.

[14] J. Beringer and E. Hüllermeier. Online clustering of parallel data streams. *Data Knowledge Engineering*, 58(2):180–204, 2006.

[15] A. Bialkowski, S. Denman, S. Sridharan, C. Fookes, and P. Lucey. A database for person re-identification in multi-camera surveillance networks. In *2012 International Conference on Digital Image Computing Techniques and Applications, DICTA 2012, Fremantle, Australia, December 3-5, 2012*, pages 1–8, 2012.

[16] A. Bifet, G. Holmes, B. Pfahringer, R. Kirkby, and R. Gavaldà. New ensemble methods for evolving data streams. pages 139–148, 2009.

[17] C. M. Bishop. *Pattern Recognition and Machine Learning (Information Science and Statistics)*. Springer-Verlag New York, Inc., Secaucus, NJ, USA, 2006.

[18] R. Capo, K. B. Dyer, and R. Polikar. Active learning in nonstationary environments. In *IJCNN*, pages 1–8, 2013.

[19] J. S. Cardoso and L. Corte-Real. Toward a generic evaluation of image segmentation. *IEEE Transactions on Image Processing*, 14:1773–1782, november 2005.

[20] A. Carlson, J. Betteridge, B. Kisiel, B. Settles, E. R. H. Jr., and T. M. Mitchell. Toward an architecture for never-ending language learning. In *Proceedings of the Twenty-Fourth AAAI Conference on Artificial Intelligence, AAAI 2010, Atlanta, Georgia, USA, July 11-15, 2010*, 2010.

[21] G. C. Cawley. Baseline methods for active learning. In *Active Learning and Experimental Design@ AISTATS*, pages 47–57, 2011.

[22] C.-C. Chang and C.-J. Lin. LIBSVM: A library for support vector machines. *ACM Transactions on Intelligent Systems and Technology*, 2:27:1–27:27, 2011.

[23] K. Chatfield, V. S. Lempitsky, A. Vedaldi, and A. Zisserman. The devil is in the details: an evaluation of recent feature encoding methods. In *BMVC*, pages 1–12, 2011.

[24] K. Chatfield, K. Simonyan, A. Vedaldi, and A. Zisserman. Return of the devil in the details: Delving deep into convolutional nets. In *British Machine Vision Conference, BMVC 2014, Nottingham, UK, September 1-5, 2014*, 2014.

[25] K. Chatfield, K. Simonyan, A. Vedaldi, and A. Zisserman. Return of the devil in the details: Delving deep into convolutional nets. In *BMVC*, 2014.

[26] C. Chen, R. Jafari, and N. Kehtarnavaz. Improving human action recognition using fusion of depth camera and inertial sensors. *IEEE T. Human-Machine Systems*, 45(1):51–61, 2015.

[27] L. Chen, L. Zou, and L. Tu. A clustering algorithm for multiple data streams based on spectral component similarity. *Information Sciences*, 183(1):35–47, 2012.

[28] X. Chen, A. Shrivastava, and A. Gupta. Neil: Extracting visual knowledge from web data. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 1409–1416, 2013.

[29] Y. Chen. Clustering parallel data streams. *Data Mining and Knowledge Discovery in Real Life Applications*, I-Tech Education and Publishing, 2009.

[30] J. M. Colores-Vargas, M. S. García-Vázquez, A. A. Ramírez-Acosta, H. Pérez-Meana, and M. Nakano-Miyatake. Video images fusion to improve iris recognition accuracy in unconstrained environments. In *5th Mexican Conference on Pattern Recognition*, pages 114–125, 2013.

[31] G. Csurka, C. R. Dance, L. Fan, J. Willamowski, and C. Bray. Visual categorization with bags of keypoints. In *In Workshop on Statistical Learning in Computer Vision, ECCV*, pages 1–22, 2004.

[32] M. Culver, K. Deng, and S. D. Scott. Active learning to maximize area under the ROC curve. In *Proceedings of the 6th IEEE International Conference on Data Mining (ICDM 2006), 18-22 December 2006, Hong Kong, China*, pages 149–158, 2006.

[33] N. Cvejic, S. Nikolov, H. Knowles, A. Loza, A. Achim, D. Bull, and C. Canagarajah. The effect of pixel-level fusion on object tracking in multi-sensor surveillance video. In *ICVPR*, pages 1–7, 2007.

[34] D. Dai, Y. Wang, Y. Chen, and L. Van Gool. Is image super-resolution helpful for other vision tasks? In *IEEE Winter Conference on Applications of Computer Vision (WACV)*, 2016.

[35] N. Dalal. *Finding People in Images and Videos*. These, Institut National Polytechnique de Grenoble - INPG, July 2006.

[36] A. P. Dempster, N. M. Laird, and D. B. Rubin. Maximum likelihood from incomplete data via the em algorithm. *Journal of the Royal Statistical Society. Series B (Methodological)*, 39(1):pp. 1–38, 1977.

[37] J. Demšar. Statistical comparisons of classifiers over multiple data sets. *Journal of Machine Learning Research*, 7:1–30, Dec. 2006.

[38] C. Dietrich, G. Palm, and F. Schwenker. Decision templates for the classification of bioacoustic time series. *Information Fusion*, 4(2):101–109, 2003.

[39] G. Ditzler and R. Polikar. Semi-supervised learning in nonstationary environments. In *IJCNN*, pages 2741–2748, 2011.

[40] G. Ditzler and R. Polikar. Incremental learning of concept drift from streaming imbalanced data. *IEEE Trans. Knowl. Data Eng.*, 25(10):2283–2301, 2013.

[41] G. Ditzler and R. Polikar. Incremental learning of concept drift from streaming imbalanced data. *IEEE Transactions on Knowledge and Data Engineering*, 25(10):2283–2301, 2013.

[42] G. Ditzler, M. Roveri, C. Alippi, and R. Polikar. Learning in nonstationary environments: A survey. *IEEE Comp. Int. Mag.*, 10(4):12–25, 2015.

[43] K. B. Dyer, R. Capo, and R. Polikar. Compose: A semisupervised learning framework for initially labeled nonstationary streaming data. *IEEE Trans. Neural Netw. Learning Syst.*, 25(1):12–26, 2014.

[44] R. Elwell and R. Polikar. Incremental learning of concept drift in nonstationary environments. *IEEE Transactions on Neural Networks*, 22(10):1517–1531, 2011.

[45] R.-E. Fan, K.-W. Chang, C.-J. Hsieh, X.-R. Wang, and C.-J. Lin. Liblinear: A library for large linear classification. *Journal of Machine Learning Research*, 9:1871–1874, 2008.

[46] M. Farenzena, L. Bazzani, A. Perina, V. Murino, and M. Cristani. Person re-identification by symmetry-driven accumulation of local features. In *The Twenty-Third IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2010, San Francisco, CA, USA, 13-18 June 2010*, pages 2360–2367, 2010.

[47] J. M. Ferryman. Pets2009 dataset, 2009.

[48] D. Figueira, M. Taiana, A. M. Nambiar, J. C. Nascimento, and A. Bernardino. The HDA+ data set for research on fully automated re-identification systems. In *Computer Vision - ECCV 2014 Workshops - Zurich, Switzerland, September 6-7 and 12, 2014, Proceedings, Part III*, pages 241–255, 2014.

[49] J. W. Fisher and T. Darrell. Signal level fusion for multimodal perceptual user interface. In *workshop on Perceptive user interfaces*, pages 1–7, 2001.

[50] J. Gama, P. Medas, G. Castillo, and P. P. Rodrigues. Learning with drift detection. In *Advances in Artificial Intelligence - SBIA 2004, 17th Brazilian Symposium on Artificial Intelligence, São Luis, Maranhão, Brazil, September 29 - October 1, 2004, Proceedings*, pages 286–295, 2004.

[51] J. Gama, R. Sebastião, and P. P. Rodrigues. On evaluating stream learning algorithms. *Machine Learning*, 90(3):317–346, 2013.

[52] E. Gavves, T. Mensink, T. Tommasi, C. G. M. Snoek, and T. Tuytelaars. Active transfer learning with zero-shot priors: Reusing past datasets for future tasks. *CoRR*, abs/1510.01544, 2015.

[53] J. Goldberger and H. Aronowitz. A distance measure between gmms based on the unscented transform and its application to speaker recognition. In *INTERSPEECH 2005 - Eurospeech, 9th European Conference on Speech Communication and Technology, Lisbon, Portugal, September 4-8, 2005*, pages 1985–1988, 2005.

[54] J. Gómez-Romero, M. A. Serrano, J. García, J. M. Molina, and G. L. Rogova. Context-based multi-level information fusion for harbor surveillance. *Information Fusion*, 21:173–186, 2015.

[55] S. Gong, M. Cristani, S. Yan, and C. C. Loy, editors. *Person Re-Identification*. Advances in Computer Vision and Pattern Recognition. Springer, 2014.

[56] S. Gupta, B. Hariharan, and J. Malik. Exploring person context and local scene context for object detection. *CoRR*, abs/1511.08177, 2015.

[57] R. Hamid, R. K. Kumar, J. K. Hodgins, and I. A. Essa. A visualization framework for team sports captured using multiple static cameras. *Computer Vision and Image Understanding*, 118:171–183, 2014.

[58] Y. Hao, Y. Chen, J. Zakaria, B. Hu, T. Rakthanmanon, and E. Keogh. Towards never-ending learning from time series streams. In *Proceedings of the 19th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, KDD '13, pages 874–882, 2013.

[59] M. Hasan and A. K. Roy-Chowdhury. Incremental activity modeling and recognition in streaming videos. In *2014 IEEE Conference on Computer Vision and Pattern Recognition*, pages 796–803, June 2014.

[60] G. Heitz and D. Koller. Learning spatial context: Using stuff to find things. In *Computer Vision - ECCV 2008, 10th European Conference on Computer Vision, Marseille, France, October 12-18, 2008, Proceedings, Part I*, pages 30–43, 2008.

[61] C. Hennig. Methods for merging gaussian mixture components. *Adv. Data Analysis and Classification*, 4(1):3–34, 2010.

[62] M. Hirzer, P. M. Roth, M. Köstinger, and H. Bischof. Relaxed pairwise learned metric for person re-identification. In *Computer Vision - ECCV 2012 - 12th European Conference on Computer Vision, Florence, Italy, October 7-13, 2012, Proceedings, Part VI*, pages 780–793, 2012.

[63] T. hsun Chang and S. Gong. Tracking multiple people with a multi-camera system. In *IEEE Workshop on Multi-Object Tracking*, 2001.

[64] O. Javed. Appearance modeling for tracking in multiple non-overlapping cameras. In *In IEEE International Conference on Computer Vision and Pattern Recognition*, pages 26–33, 2005.

[65] O. Javed and M. Shah. *Automated Multi-Camera Surveillance: Algorithms and Practice*, volume 10 of *The International Series in Video Computing*. Springer, 2008.

[66] B. Jiang, B. Martínez, M. F. Valstar, and M. Pantic. Decision level fusion of domain specific regions for facial action recognition. In *22nd International Conference on Pattern Recognition, ICPR 2014, Stockholm, Sweden, August 24-28, 2014*, pages 1776–1781, 2014.

[67] H. Josinski, A. Michalczuk, D. Kostrzewa, A. Switonski, and K. W. Wojciechowski. Heuristic method of feature selection for person re-identification based on gait motion capture data. In *Intelligent Information and Database Systems - 6th Asian Conference, ACIIDS 2014, Bangkok, Thailand, April 7-9, 2014, Proceedings, Part II*, pages 585–594, 2014.

[68] Y. Kamishima, N. Inoue, and K. Shinoda. *EURASIP J. Image and Video Processing*, 2013.

[69] S. Karanam, Y. Li, and R. J. Radke. Person re-identification with discriminatively trained viewpoint invariant dictionaries. In *2015 IEEE International Conference on Computer Vision, ICCV 2015,Santiago, Chile*, pages 4516–4524, 2015.

[70] E. Keogh and S. Kasetty. On the need for time series data mining benchmarks: A survey and empirical demonstration. *Data Mining Knowledge Discovery*, 7(4):349–371, Oct. 2003.

[71] S. S. Khan and M. G.Madden. A survey of recent trends in one class classification. In L. Coyle and J. Freyne, editors, *Artificial Intelligence and Cognitive Science*, volume 6206 of *Lecture Notes in Computer Science*, pages 188–197. Springer Berlin Heidelberg, 2010.

[72] A. Khoshrou, A. P. Aguiar, and F. L. Pereira. Adaptive sampling using an unsupervised learning of gmms applied to a fleet of auvs with ctd measurements. In *Robot 2015: Second Iberian Robotics Conference*, pages 321–332. Springer, 2016.

[73] S. Khoshrou, J. S. Cardoso, E. Granger, and L. F. Teixeira. Spatio-temporal fusion for learning of regions of interests over multiple video streams. In *Advances in Visual Computing - 11th International Symposium, ISVC 2015, Las Vegas, NV, USA, December 14-16, 2015, Proceedings, Part II*, pages 509–520, 2015.

[74] S. Khoshrou, J. S. Cardoso, and L. F. Teixeira. Active learning from video streams in a multi-camera scenario. In *22nd International Conference on Pattern Recognition, ICPR 2014, Stockholm, Sweden, August 24-28, 2014*, pages 1248–1253, 2014.

[75] S. Khoshrou, J. S. Cardoso, and L. F. Teixeira. Active mining of parallel video streams. *CoRR*, abs/1405.3382, 2014.

[76] S. Khoshrou, J. S. Cardoso, and L. F. Teixeira. Learning from evolving video streams in a multi-camera scenario. *Machine Learning*, 100(2-3):609–633, 2015.

[77] J. Kittler, M. Hatef, R. P. W. Duin, and J. Matas. On combining classifiers. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 20(3):226–239, 1998.

[78] A. Klaser, M. Marszałek, and C. Schmid. A spatio-temporal descriptor based on 3d-gradients. In *BMVC 2008-19th British Machine Vision Conference*, pages 275–1. British Machine Vision Association, 2008.

[79] J. Z. Kolter and M. A. Maloof. Dynamic weighted majority: An ensemble method for drifting concepts. *Journal of Machine Learning Research*, 8:2755–2790, 2007.

[80] C. Krishna Mohan, N. Dhananjaya, and B. Yegnanarayana. Video shot segmentation using late fusion technique. In *ICMLA*, pages 267–270, Dec 2008.

[81] H. W. Kuhn. The Hungarian Method for the Assignment Problem. *Naval Research Logistics Quarterly*, 2(1–2):83–97, March 1955.

[82] C.-H. Kuo, C. Huang, and R. Nevatia. Inter-camera association of multi-target tracks by online learned appearance affinity models. In *Proceedings of the 11th European Conference on Computer Vision: Part I*, ECCV'10, pages 383–396, 2010.

[83] T. H. Lam, K. H. Cheung, and J. N. Liu. Gait flow image: A silhouette-based gait representation for human identification. *Pattern recognition*, 44(4):973–987, 2011.

[84] Y. LeCun, B. Boser, J. S. Denker, D. Henderson, R. E. Howard, W. Hubbard, and L. D. Jackel. Backpropagation applied to handwritten zip code recognition. *Neural Comput.*, 1(4):541–551, Dec. 1989.

[85] D. Li, L. Xu, and E. Goodman. On-line em variants for multivariate normal mixture model in background learning and moving foreground detection. *Journal of mathematical imaging and vision*, 48(1):114–133, 2014.

[86] P. Li, X. Wu, and X. Hu. Mining recurring concept drifts with limited labeled streaming data. *ACM Trans. Intell. Syst. Technol.*, 3(2):29:1–29:32, Feb. 2012.

[87] S.-N. Lim, L. S. Davis, and A. Elgammal. A scalable image-based multi-camera visual surveillance system. In *Proceedings of the IEEE Conference on Advanced Video and Signal Based Surveillance*, AVSS '03, pages 205–212, Washington, DC, USA, 2003. IEEE Computer Society.

[88] K. Liu, B. Ma, W. Zhang, and R. Huang. A spatio-temporal appearance representation for viceo-based pedestrian re-identification. In *The IEEE International Conference on Computer Vision (ICCV)*, December 2015.

[89] D. G. Lowe. Object recognition from local scale-invariant features. In *Proceedings of the International Conference on Computer Vision*, ICCV '99, pages 1150–1157, 1999.

[90] Z. Ma, Q. Dai, and N. Liu. Several novel evaluation measures for rank-based ensemble pruning with applications to time series prediction. *Expert Systems with Applications*, 42(1):280–292, 2015.

[91] M. Markou and S. Singh. Novelty detection: a review–part 1: statistical approaches. *Signal Processing*, 83(12):2481–2497, Dec. 2003.

[92] N. Martinel, C. Micheloni, and G. L. Foresti. A pool of multiple person re-identification experts. *Pattern Recogn. Lett.*, 71(C):23–30, Feb. 2016.

[93] N. Martinel, C. Micheloni, and C. Piciarelli. Learning pairwise feature dissimilarities for person re-identification. In *Seventh International Conference on Distributed Smart Cameras, ICDSC 2013, October 29 2013-November 1, 2013, Palm Springs, CA, USA*, pages 1–6, 2013.

[94] G. Martínez-Muñoz, D. Hernández-Lobato, and A. Suárez. An analysis of ensemble pruning techniques based on ordered aggregation. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 31(2):245–259, 2009.

[95] M. M. Masud, Q. Chen, L. Khan, C. Aggarwal, J. Gao, J. Han, and B. Thuraisingham. Addressing concept-evolution in concept-drifting data streams. In *Proceedings of the 2010 IEEE International Conference on Data Mining*, ICDM '10, pages 929–934, Washington, DC, USA, 2010. IEEE Computer Society.

[96] M. M. Masud, J. Gao, L. Khan, J. Han, and B. M. Thuraisingham. Classification and novel class detection in data streams with active mining. In *PAKDD (2)*, pages 311–324, 2010.

[97] M. M. Masud, J. Gao, L. Khan, J. Han, and B. M. Thuraisingham. Classification and novel class detection in concept-drifting data streams under time constraints. *IEEE Transactions on Knowledge Data Engineering*, 23(6):859–874, 2011.

[98] B. C. Matei, H. S. Sawhney, and S. Samarasekera. Vehicle tracking across nonoverlapping cameras using joint kinematic and appearance features. CVPR '11, pages 3465–3472, 2011.

[99] R. Mazzon, S. F. Tahir, and A. Cavallaro. Person re-identification in crowd. *Pattern Recogn. Lett.*, 33(14):1828–1837, Oct. 2012.

[100] H. Moon and P. J. Phillips. Computational and performance aspects of pca-based face-recognition algorithms. *Perception*, 30(3):303–321, 2001.

[101] C. Nakajima, M. Pontil, B. Heisele, and T. A. Poggio. Full-body person recognition system. *Pattern Recognition*, 36(9):1997–2006, 2003.

[102] G. H. Nguyen, A. Bouzerdoum, and S. L. Phung. *Learning Pattern Classification Tasks with Imbalanced Data Sets*. Intech, 2009.

[103] M. S. Nixon, P. L. Correia, K. Nasrollahi, T. B. Moeslund, A. Hadid, and M. Tistarelli. On soft biometrics. *Pattern Recognition Letters*, 68:218–230, 2015.

[104] M. S. Nixon, T. Tan, and R. Chellappa. *Human identification based on gait*, volume 4. Springer Science & Business Media, 2010.

[105] T. Ojala, M. Pietikäinen, and D. Harwood. A comparative study of texture measures with classification based on featured distributions. *Pattern Recognition*, 29(1):51–59, 1996.

[106] M. Oren, C. Papageorgiou, P. Sinha, E. Osuna, and T. A. Poggio. Pedestrian detection using wavelet templates. In *1997 Conference on Computer Vision and Pattern Recognition (CVPR '97), June 17-19, 1997, San Juan, Puerto Rico*, pages 193–199, 1997.

[107] S. Ozawa, S. L. Toh, S. Abe, S. Pang, and N. Kasabov. Incremental learning of feature space and classifier for face recognition. *Neural Networks*, 18(5-6):575–584, 2005.

[108] C. C. Pagano, E. Granger, R. Sabourin, G. L. Marcialis, and F. Roli. Adaptive ensembles for face recognition in changing video surveillance environments. *Inf. Sci.*, 286:75–101, 2014.

[109] F. Perronnin, J. Sánchez, and T. Mensink. Improving the fisher kernel for large-scale image classification. In *Proceedings of the 11th European Conference on Computer Vision: Part IV*, ECCV'10, pages 143–156, Berlin, Heidelberg, 2010. Springer-Verlag.

[110] R. Pflugfelder and H. Bischof. Localization and trajectory reconstruction in surveillance cameras with nonoverlapping views. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 32(4):709–721, 2010.

[111] D. Povey, S. M. Chu, and B. Varadarajan. Universal background model based speech recognition. In *IEEE International Conference on Acoustics, Speech and Signal Processing*, pages 4561–4564, 2008.

[112] C. project consortium. Caviar dataset, 2001.

[113] D. Reynolds. Gaussian mixture models. *Encyclopedia of Biometric Recognition*, pages 12–17, 2008.

[114] D. Reynolds, T. Quatieri, and R. Dunn. Speaker verification using adapted gaussian mixture models. *Digital signal processing*, 10(1):19–41, 2000.

[115] D. A. Reynolds. An overview of automatic speaker recognition technology. In *Acoustics, Speech, and Signal Processing (ICASSP), 2002 IEEE International Conference on*, volume 4, pages IV–4072. IEEE, 2002.

[116] P. P. Rodrigues, J. Gama, and J. P. Pedroso. Hierarchical clustering of time-series data streams. *IEEE Transaction on Knowledge Data Engineering*, 20(5):615–627, 2008.

[117] A. Runnalls. Kullback-leibler approach to gaussian mixture reduction. *IEEE Transactions on Aerospace and Electronic Systems*, 3(43):989–999, 2007.

[118] J. Sánchez, F. Perronnin, T. Mensink, and J. J. Verbeek. Image classification with the fisher vector: Theory and practice. *International Journal of Computer Vision*, 105(3):222–245, 2013.

[119] D. Schacter, D. T. Gilbert, and D. M. Wegner. *Psychology (2nd Edition)*. Worth, New York, 2011.

[120] W. J. Scheirer, L. P. Jain, and T. E. Boult. Probability models for open set recognition. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 36(11):2317–2324, Nov 2014.

[121] M. Schels, M. Glodek, S. Meudt, S. Scherer, M. Schmidt, G. Layher, S. Tschechne, T. Brosch, D. Hrabal, S. Walter, H. C. Traue, G. Palm, F. Schwenker, M. Rojc, and N. Campbell. Multi-Modal Classifier-Fusion for the Recognition of Emotions. In *Converbal Synchrony in Human-Machine Interaction*, pages 73–97. CRC Press, 2013.

[122] H. Sellahewa and S. A. Jassim. Image-quality-based adaptive face recognition. *IEEE Transactions on Instrumentation and Measurement*, 59(4):805–813, 2010.

[123] B. Settles. Active learning literature survey. Technical Report 1648, University of Wisconsin–Madison, 2009.

[124] Y. Shan, H. S. Sawhney, and R. Kumar. Unsupervised learning of discriminative edge measures for vehicle matching between non-overlapping cameras. In *CVPR (1)*, pages 894–901, 2005.

[125] V. Sharma and J. W. Davis. Feature-level fusion for object segmentation using mutual information. In *Augmented Vision Perception in Infrared*, pages 295–320. Springer, 2009.

[126] K. Shinoda and N. Inoue. Reusing speech techniques for video semantic indexing [applications corner]. *Signal Processing Magazine, IEEE*, 30(2):118–122, 2013.

[127] D. Simonnet, M. Lewandowski, S. A. Velastin, J. Orwell, and E. Turkbeyler. Re-identification of pedestrians in crowds using dynamic time warping. In *Proceedings of the 12th International Conference on Computer Vision - Volume Part I*, ECCV'12, pages 423–432, Berlin, Heidelberg, 2012. Springer-Verlag.

[128] S. G. Soares and R. Araújo. An adaptive ensemble of on-line extreme learning machines with variable forgetting factor for dynamic system prediction. *Neurocomputing*, 171:693–707, 2016.

[129] W. N. Street and Y. Kim. A streaming ensemble algorithm (sea) for large-scale classification. In *Proceedings of the Seventh ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, KDD '01, pages 377–382, New York, NY, USA, 2001. ACM.

[130] C. Su and A. Amer. A real-time adaptive thresholding for video change detection. In *Proceedings of the International Conference on Image Processing, ICIP 2006, October 8-11, Atlanta, Georgia, USA*, pages 157–160, 2006.

[131] J. S. Supancic III and D. Ramanan. Self-paced learning for long-term tracking. In *Proceedings of the 2013 IEEE Conference on Computer Vision and Pattern Recognition*, CVPR '13, pages 2379–2386, Washington, DC, USA, 2013. IEEE Computer Society.

[132] M. Taiana, D. Figueira, A. Nambiar, J. Nascimento, and A. Bernardino. Towards fully automated person re-identification. In *Computer Vision Theory and Applications (VISAPP), 2014 International Conference on*, volume 3, pages 140–147, Jan 2014.

[133] G. Takacs, V. Chandrasekhar, S. Tsai, D. Chen, R. Grzeszczuk, and B. Girod. Unified real-time tracking and recognition with rotation-invariant fast features. In *Proceedings of IEEE Conference on Computer Vision And Pattern Recognition (CVPR'10)*, 2010.

[134] Q. Tao and R. Veldhuis. Hybrid fusion for biometrics: Combining score-level and decision-level fusion. In *2008 IEEE Computer Society Conference on Computer Vision and Pattern Recognition, Workshop on Biometrics*, pages 1–6, Los Alamitos, June 2008. IEEE Computer Society Press.

[135] L. F. Teixeira, P. Carvalho, J. S. Cardoso, and L. Corte-Real. Automatic description of object appearances in a wide-area surveillance scenario. In *19th IEEE International Conference on Image Processing*, pages 1609–1612, 2012.

[136] L. F. Teixeira and L. Corte-Real. Video object matching across multiple independent views using local descriptors and adaptive learning. *Pattern Recognition Letters*, 30(2):157–167, 2009.

[137] E. Tola, V. Lepetit, and P. Fua. Daisy: An efficient dense descriptor applied to wide baseline stereo. *IEEE TRANS. PATTERN ANALYSIS AND MACHINE INTELLIGENCE*, 32(5), 2010.

[138] N. Ueda, R. Nakano, Z. Ghahramani, and G. E. Hinton. Split and merge EM algorithm for improving gaussian mixture density estimates. *VLSI Signal Processing*, 26(1-2):133–140, 2000.

[139] R. Vezzani, D. Baltieri, and R. Cucchiara. People reidentification in surveillance and forensics: A survey. *ACM Comput. Surv.*, 46(2):29, 2013.

[140] H. Wang, D. Oneata, J. Verbeek, and C. Schmid. A robust and efficient video representation for action recognition. *International Journal of Computer Vision*, pages 1–20.

[141] T. Wang, S. Gong, X. Zhu, and S. Wang. Person re-identification by discriminative selection in video ranking. *IEEE Trans. Pattern Anal. Mach. Intell.*, abs/1601.06260, 2016.

[142] X. Wang. Intelligent multi-camera video surveillance: A review. *Pattern Recognition Letters*, 34(1):3–19, 2013.

[143] L. Wei, Y. Tian, Y. Wang, and T. Huang. Swiss-system based cascade ranking for gait-based person re-identification. In *AAAI*, pages 1882–1888, 2015.

[144] J. Wu. An online-optimized incremental learning framework for video semantic classification. In *12th ACM International Conference on Multimedia*, pages 320–323, 2004.

[145] Z. Wu, Y. Li, and R. J. Radke. Viewpoint invariant human re-identification in camera networks using pose priors and subject-discriminative features. *IEEE Trans. Pattern Anal. Mach. Intell.*, 37(5):1095–1108, 2015.

[146] Z. Xiong, T. Zheng, Z. Song, F. Soong, and W. Wu. A tree-based kernel selection approach to efficient gaussian mixture model–universal background model based speaker identification. *Speech communication*, 48(10):1273–1282, 2006.

[147] X. Yang and Q. Tian. Video repeat recognition and mining by visual features. In *Video Search and Mining*, pages 305–326. 2010.

[148] D. F. Yuyin Sun. Neol: Toward never-ending object learning for robots. IEEE International Conference on Robotics and Automation (ICRA), 2016.

[149] K. Zhang, L. Zhang, Q. Liu, D. Zhang, and M. Yang. Fast visual tracking via dense spatio-temporal context learning. In *Computer Vision - ECCV 2014 - 13th European Conference, Zurich, Switzerland, September 6-12, 2014, Proceedings, Part V*, pages 127–141, 2014.

[150] Y. Zhang. *Constructive and Destructive Optimization Methods for Predictive Ensemble Learning*. ProQuest, 2006.

[151] Y. Zhang, S. Burer, and W. N. Street. Ensemble pruning via semi-definite programming. *J. Mach. Learn. Res.*, 7:1315–1338, Dec. 2006.

[152] L. Zheng, L. Shen, L. Tian, S. Wang, J. Wang, and Q. Tian. Scalable person re-identification: A benchmark. In *2015 IEEE International Conference on Computer Vision, ICCV 2015, Santiago, Chile, December 7-13, 2015*, pages 1116–1124, 2015.

[153] W. Zheng, S. Gong, and T. Xiang. Towards open-world person re-identification by one-shot group-based verification. *IEEE Trans. Pattern Anal. Mach. Intell.*, 38(3):591–606, 2016.

[154] W.-S. Zheng, S. Gong, and T. Xiang. Person re-identification by probabilistic relative distance comparison. In *CVPR*, pages 649–656, 2011.

[155] Z.-H. Zhou. Ensemble methods: Foundations and algorithms. 1st. Chapman & Hall/CRC, 2012.

[156] I. Zliobaite, A. Bifet, B. Pfahringer, and G. Holmes. Active learning with evolving streaming data. In D. Gunopulos, T. Hofmann, D. Malerba, and M. Vazirgiannis, editors, *ECML/PKDD (3)*, volume 6913 of *Lecture Notes in Computer Science*, pages 597–612. Springer, 2011.

[157] I. Zliobaite, A. Bifet, B. Pfahringer, and G. Holmes. Active learning with drifting streaming data. *IEEE Trans. Neural Netw. Learning Syst.*, 25(1):27–39, 2014.

[158] C. Zor, T. Windeatt, and J. Kittler. ECOC matrix pruning using accuracy information. In *Multiple Classifier Systems, 11th International Workshop, MCS 2013, Nanjing, China, May 15-17, 2013. Proceedings*, pages 386–397, 2013.